

**Tiling Euclidean Polygons Mapped From Their Hyperbolic
Equivalent**

**A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE
SCHOOL OF THE UNIVERSITY OF MINNESOTA
BY**

Manoj Naik Prakash

**IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE**

Dr. Douglas Dunham

June 2017

© Manoj Naik Prakash 2017

Acknowledgements

I would like to express my deep gratitude for my faculty advisor, Dr. Douglas Dunham's constant support and patience throughout the course of my work. His guidance at every step has allowed my ideas to take shape and has helped me realize my goals.

I would also like to thank Dr. Pete Willemsen for his incessant encouragement and motivation, and Dr. Bruce Peckham for his valuable time and consideration in being my committee member. I am immensely thankful to the faculty of the Computer Science Department who have moulded my career and vastly contributed to my skill-set & learning experience. I would also like to acknowledge the CS office staff, Lori Lucia and Clare Ford and greatly appreciate their valuable assistance through and through.

Last but not the least, I would like to thank all the friends I made in Duluth who I share beautiful memories with. I takeaway with me a journey that I would always cherish.

Dedication

Dedicated to my mom, Mrs. Rukmini Naik, my dad, Mr. Prakash Naik, and my sister Megha Naik, for their unconditional love, and unwavering faith in me and for being a support system that never fails.

Abstract

The concept of repeating artistic patterns, for instance spirals, waves, snail shells, tilings etc., have been in existence for centuries now. It was during 1900's that a noted Dutch graphic artist M.C. Escher worked extensively in this world of art which was inspired by mathematics. Escher painstakingly hand-drew such perceptive repeating patterns (which were mostly Euclidean in nature) and his famous hyperbolic patterns: *Circle Limit I, II, III and IV* which were based on regular tessellations.

This research work concentrates on leveraging hyperbolic and Euclidean geometry in art, drawing inspiration from Escher's work. Various Euclidean, non-Euclidean and spherical repeating patterns are special forms of tessellations. At the core of these patterns lies an idea, proposed by Dr. Dunham, that a small congruent sub-pattern, called a motif, which when reflected and rotated will generate the entire pattern. This work focuses on transforming the central polygon in a hyperbolic pattern to obtain its Euclidean counterpart. This counterpart will further tile a planar region to generate a Euclidean tiling. There are various interesting applications that allow the user to draw such repeating patterns programmatically, however none of them show the reverse mapping from a hyperbolic to a Euclidean pattern. We enhance an existing Java application by creating a bridge that connects hyperbolic patterns to their Euclidean equivalents and facilitates the user to work with tilings. The results are expected to show a transformation from hyperbolic to Euclidean patterns followed by tiling of the Euclidean pattern on a planar region.

Contents

List of Figures	vi
1 Introduction.....	1
2 Kinds of Geometry	5
2.1 Euclidean Geometry	5
2.2 Non - Euclidean Geometry	7
2.2.1 Hyperbolic Geometry	8
2.2.2 Elliptic Geometry.....	9
3 Hyperbolic Geometry Models.....	12
3.1 Poincaré Disk Model	12
3.2 Klein Model.....	15
3.3 Weierstrass Model	18
3.4 Isomorphisms.....	19
3.4.1 Weierstrass & Poincaré Disk Model Isomorphism.....	20
3.4.2 Weierstrass & Klein Model Isomorphism	20
3.4.3 Poincaré Disk & Klein Model Isomorphism	21
4 Euclidean Patterns	23
4.1 Tessellation.....	23
4.2 Fundamental Region and Motif	24
4.3 Wallpaper Group	24
5 Implementation	25
5.1 Transforming Poincaré Model Points To Klein Model	25

5.2	Generation Of Central Euclidean Pattern Algorithm	30
5.3	Tiling Euclidean polygons.....	36
5.3.1	Tiling Euclidean Polygon Of Type: Equilateral Triangle.....	36
5.3.2	Tiling Euclidean Polygon Of Type: Square.....	44
5.3.3	Tiling Euclidean Polygon Of Type: Regular Hexagon.....	47
6	Graphical User Interface.....	50
7	Results	55
8	Conclusion	70
9	Future Work	71
	References.....	72
	Appendix A.....	73

List of Figures

1.1 Central polygon, a $\{4, 6\}$ tessellation in its Hyperbolic form	3
1.2 Euclidean equivalent of the above polygon	3
1.3 Tiling the $\{4, 4\}$ Euclidean polygon on a planar region	4
2.1 Parallel postulate in Euclidean Geometry	6
2.2 Parallel postulate in Hyperbolic Geometry	8
2.3 Figure with a positive curvature.....	10
2.4 Circles representing lines in Elliptic geometry. Colored dots represent antipodal points.....	10
3.1 Lines in a Poincaré disk model	13
3.2 A square in the Poincaré model (the red strips are part of a design pattern).....	14
3.3 A triangle in the Poincaré model (the red strips are part of a design pattern).....	14
3.4 Lines(or open chords) in Klein model	16
3.5 A square in Klein model	17
3.6 A triangle in the Klein model.....	17
3.7 Every line in a Weierstrass model is a hyperbola	19
5.1 The central Euclidean triangle divided into three isosceles triangles of similar size (indicated in brown).....	26
5.2 The central Euclidean square divided into four isosceles triangles of similar size (indicated in brown).....	26
5.3 The central Euclidean hexagon divided into six equilateral triangles of similar size (indicated in brown)	27
5.4 Partially completed central Euclidean square, composed of three line segments, after the 1st iteration. This indicates the starting figure.....	29

5.5 Partially completed central Euclidean square after the 2nd iteration.....	29
5.6 Partially completed central Euclidean square after the 3rd iteration	30
5.7 Fully completed central Euclidean square after the final(4th) iteration.....	30
5.8 Rotation (only) of motif about the origin starting from step (a) until (d) to create the central polygon.	32
5.9 Reflection across the edge bisector. (a) Initial motif created. (b) Reflection of actual motif created in step a across the edge. (c) Motif created in step b is rotated about the origin. (d) Motif created in step b is rotated about the origin once again to complete the central polygon.	33
5.10 Reflection across a hypotenuse. (a) Initial motif created (b) Reflection of motif created in step a about its hypotenuse. (c) Final polygon created by rotating motif created in step (b)....	34
5.11 {3, 6} Euclidean triangle.....	37
5.12 Rotated {3, 6} Euclidean triangle	37
5.13 Euclidean triangle on an XY plane	38
5.14 Detailed intermediate result in tiling a {3, 6} triangle using the above algorithm.....	40
5.15 Inverse of the Euclidean pattern of shown in figure 5.10	40
5.16 Detailed image of the inverse of the Euclidean triangle in XY plane.....	41
5.17 Inverse tiling of {3, 6} Euclidean triangle	43
5.18 Euclidean square in XY plane.....	44
5.19 A {4, 4} Euclidean square	45
5.20 Detailed Tiling of {4, 5} Euclidean square.....	46
5.21 Detailed hexagon on an XY grid.....	47
5.22 Tiling a {6, 3} Euclidean hexagon.....	49
6.1 Opening a data file	50
6.2 The above selection prompts a dialog box for the user to choose a data file.....	51

6.3 "Generate Euclidean Equivalent" & "Generate tiling" features under "Edit" menu section..	52
(Hyperbolic pattern drawn on the canvas after the file is loaded)	52
6.4 Clicking on "Generate Euclidean Equivalent" after loading the file.....	53
6.5 Clicking on "Generate Tiling" after loading the file	54
7.1 {3,7} Triangle (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern	56
7.2 {3,7} Butterfly triangle : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern	57
7.3 {3,8} triangle : (a) In hyperbolic form (b) Euclidean counterpart (c) tiling pattern	58
7.4 {3, 24} triangle : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern.....	59
7.5 Escher circle limit I {4, 6} square : (a) in hyperbolic form (b) Euclidean counterpart.....	60
(c) tiling pattern.....	60
7.6 A special case: Escher circle limit IV {4, 5} square : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern	61
7.7 {4, 6} frog square : (a) in hyperbolic form (b) Euclidean counterpart.....	62
(c) tiling pattern.....	62
7.8 {4, 5} fish square : (a) in hyperbolic form (b) Euclidean counterpart	63
(c) tiling pattern.....	63
7.9 {4, 5} square : (a) in hyperbolic form (b) Euclidean counterpart	64
(c) tiling pattern.....	64
7.10 {4, 6} square : (a) in hyperbolic form (b) Euclidean counterpart	65
(c) tiling pattern.....	65
7.11 {6, 4} hexagon with woven ribbons : (a) in hyperbolic form (b) Euclidean counterpart	66
(c) tiling pattern.....	66
7.12 Central polygon in Coxeter's figure - {6, 4} hexagon : (a) in hyperbolic form	67

(b) Euclidean counterpart (c) tiling pattern.....	67
7.13 A $\{6, 4\}$ hexagon with scattered strips of ribbon: (a) in hyperbolic form	68
(b) Euclidean counterpart (c) tiling pattern.....	68
7.14 $\{6, 4\}$ hexagon : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern	69

1 Introduction

The concept of repeating patterns have been used for centuries now and still strongly hold their relevance. These patterns are used in various parts of the world such as Europe, China, Arabia, etc., as designs for decorating walls, 3D objects and blocks, that vastly improve the aesthetic ambience of a place. Complex geometric art is the leitmotif in Islamic art, which involves creating beautiful patterns using simple regular/irregular shapes. A repeating pattern can be defined [1] as a pattern that remains invariant under transformations of a plane. The hyperbolic patterns are represented on a hyperbolic plane which can be visualized using a Poincaré disc model.

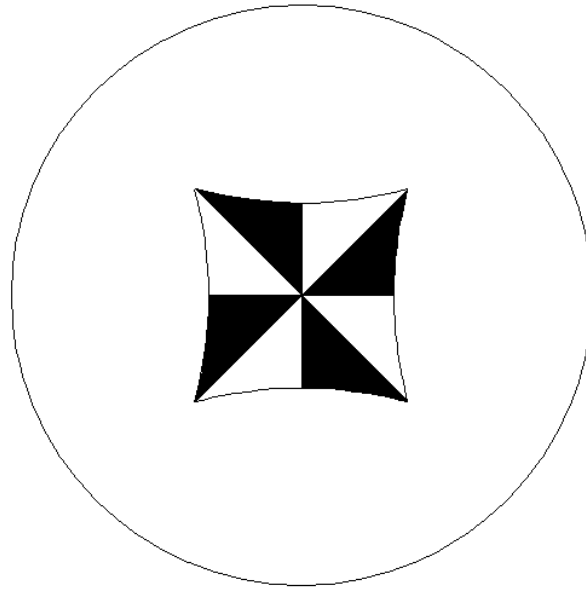
It was the noted Dutch artist, M.C. Escher who revolutionized the idea of creating mathematically inspired art. It was during his travel in the Italian country side that captured his imagination for creating interlocking repeating patterns. Despite the lack of computers during that period of time, M.C. Escher hand-drew such geometrical intricacies, which turned out the most influential and respected work in the field of Mathematics. His famous work involved lithographs, woodcuts and wood engravings which were mostly Euclidean in nature, and the hyperbolic tessellations called: *The Circle Limit I, II, III & IV*.

This research work draws inspiration from the aforementioned work and involves working with algorithms proposed by Dr. Douglas Dunham [2], for creating such repeating patterns programmatically. There are many applications

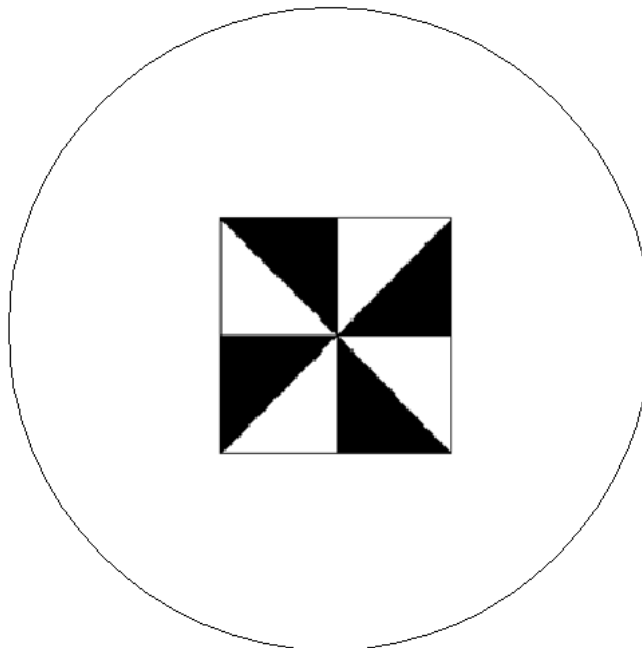
that facilitate the user in drawing repeating patterns that are hyperbolic in nature, however none of them capture the mapping from a hyperbolic pattern to its Euclidean counterpart. This project revolves around the idea of constructing a bridge from the Poincaré model, that depicts central polygon of a hyperbolic pattern, to the Klein model showing its Euclidean format. The most interesting aspect about this work is that there can be many wallpaper group patterns that eventually can be made from the Euclidean polygon by tiling it on a planar region.

A repeating pattern can also be defined as a tessellation of a plane, where tessellation means knitting small copies of a design together such that there are no overlaps or gaps. A regular tessellation is indicated by the Schläfli symbol $\{p, q\}$, which means a tiling of a plane is done by regular p -sided polygons meeting edge to edge and vertex to vertex, with q meeting at a vertex. If $(p-2)(q-2) > 4$, the tessellation is hyperbolic. Otherwise, if $(p-2)(q-2) = 4$, then the tessellation is Euclidean. The only Euclidean tessellations are $\{3, 6\}$, the tessellation by equilateral triangles, $\{4, 4\}$, the tessellation by squares, and $\{6, 3\}$, the tessellation by regular hexagons. The goal of this research is to transform hyperbolic patterns based on $\{p, q\}$ tessellation to repeating Euclidean patterns where possible. This can be done in three cases. Hyperbolic patterns based on $\{3, q\}$, $\{4, q\}$ and $\{6, q\}$ can be transformed to $\{3, 6\}$, $\{4, 4\}$ and $\{6, 3\}$ patterns respectively.

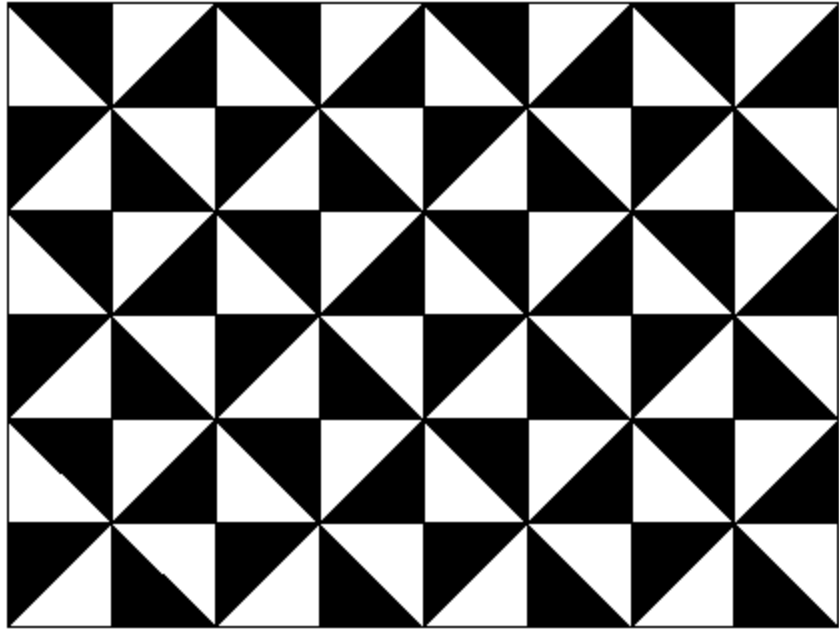
We have enhanced an existing Java application that allows the user to create a hyperbolic pattern and see its transformation to Euclidean geometry. The results are images that show the hyperbolic central polygon, its Euclidean format and finally a tiled wallpaper group image. Sample images from the application are shown below



1.1 Central polygon, a $\{4, 6\}$ tessellation in its Hyperbolic form



1.2 Euclidean equivalent of the above polygon



1.3 Tiling the $\{4, 4\}$ Euclidean polygon on a planar region

2 Kinds of Geometry

Geometry was first used in the Egypt circa 2000 B.C. It later evolved in Greece by Thales, a Greek mathematician, who introduced the concepts of edges, lines, points and positions. Geometry is a branch of mathematics which deals with the properties of geometric objects like points, curves, circles, polygons etc. *Geometry* is a Greek term, where in *geo* means earth and *metron* means measure. Geometry is medium through which we materialize the ideas of the physical world and can be used to resolve various inconsistencies.

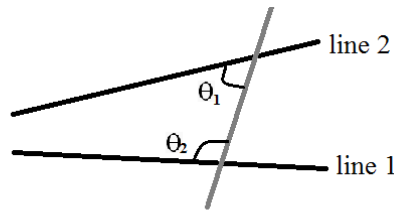
Diagrams, facts and axioms are important and indispensable tools of Geometry. For years now, geometry has been evolving and growing by addition of new procedures and concepts. There are different kinds of classical geometry that are most commonly used for the analysis of artistic patterns: (i) Euclidean (or Planar) geometry, which is used in day-to-day life and deals primarily with two dimensional lines, circles and squares, (ii) spherical geometry, which includes spherical lines and objects, (iii) hyperbolic geometry that depicts hyperbolic lines and curves.

2.1 Euclidean Geometry

Euclidean geometry is the most basic form of geometry used in every-day life for evaluating lengths, areas and volumes. Euclid's (330 B.C.) geometry is a fundamental building block in the field of Mathematics. It is the study of flat space and is also known as an axiomatic system where in all the theorems have been derived from axioms. The five important postulates (or axioms) of Euclidean

geometry are listed below [4]:

1. Between two points there can always be drawn a straight line.
2. Interior angles in a triangle sum to 180° .
3. With any point and at any distance (radius) from the point, a circle can be drawn.
4. All right angles are congruent.
5. If there are two lines intersecting another line such that the sum of the inner angles on one side is less than 180° , then the two lines invariably must meet each other on that particular side if they are stretched to an infinite length. This axiom also stands as the famous “parallel postulate”. The image below depicts this postulate:



2.1 Parallel postulate in Euclidean Geometry

Euclidean geometry was the only known geometry existing for thousands of years. Many mathematicians have regarded the fifth postulate i.e., the parallel postulate, as something that is unique and is not comparable to the four other axioms. For decades, many mathematicians tried to invalidate and negate the parallel postulate i.e., they tried to show that the parallel postulate is inconsistent with the previous four axioms. Many mathematicians tried to prove that the fifth

postulate can be rather be proved instead of calling it an axiom, but they couldn't arrive at such contradiction. However, this led to the birth of non-Euclidean geometry.

2.2 Non - Euclidean Geometry

Three mathematicians, C. F Gauss from Germany, N. Lobachevsky from Russia, and J. Bolyai from Hungary are associated with the discovery of non-Euclidean geometry. They tried to negate the fifth postulate of Euclidean geometry by trying to arrive at a conclusion that, through a point not on a line, there could be no lines or more than one line parallel to the particular line. But it was first discovered by Gauss that there could be no contradiction that can be obtained in this way. Gauss, recounted his discovery in a private note stating [5] that the assumption of sum of angles in a triangle to be less than 180° generates a curious and consistent geometry. Unfortunately, Gauss did not publish his work as he was skeptical about its acceptance in the local mathematical society. He was wary of their reaction to the negation of Euclid's geometry. However he introduced to the world, the term "Non-Euclidean" geometry.

In the 30 years following Gauss's invention of non-Euclidean geometry, in an independent study it was J. Bolyai who negated the fifth postulate and published his work on non-Euclidean geometry as a book in 1832. In another independent study, Lobachevsky published a paper on new geometry in 1829, and it was not until years after his demise that his name was associated with non-Euclidean geometry.

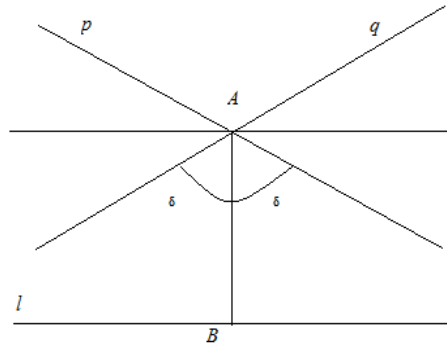
Non-Euclidean geometry is otherwise called as kinematic geometry. The two conventional branches of non-Euclidean geometry are hyperbolic geometry and elliptic geometry.

2.2.1 Hyperbolic Geometry

As a consequence to deny Euclid's fifth postulate or the parallel postulate, a new coherent and consistent geometry was invented, which stated [6]:

"Given a line l and a point P not on it, there is more than one line passing through P that is parallel to l ". Here, parallel lines means that the lines never intersect.

In a two-dimensional plane, for any line l and a point A which doesn't lie on l , there exists only one line through A which does not intersect l , whereas in Hyperbolic geometry there are two or more lines passing through A which do not intersect l as depicted in the image below :



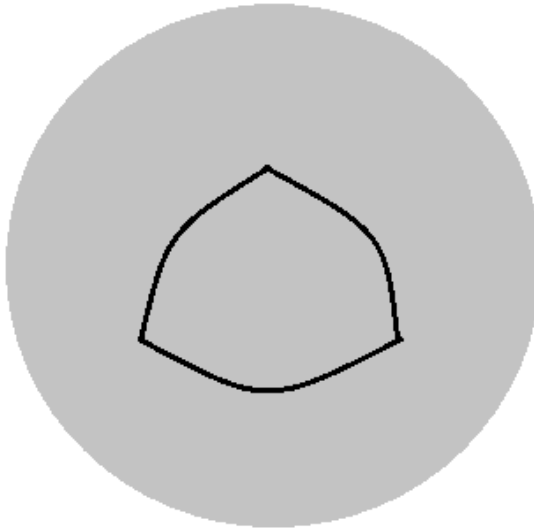
2.2 Parallel postulate in Hyperbolic Geometry

As shown in the figure, lines p and q pass through a point A , and are parallel to line B . This could be better visualized in different models used for showing Hyperbolic figures. There are three such models, namely: the Poincaré Disk, the Klein and the Weierstrass models explained in the sections ahead. Other important theorems of hyperbolic geometry are:

1. Rectangles, which is defined as four right angles, do not exist.
2. All triangles have sum of their angles less than 180° .
3. In reference to the second point, all the convex quadrilaterals have a sum of angles that is less than 360° .
4. If two triangles are similar then it implies that the triangles are congruent.

2.2.2 Elliptic Geometry

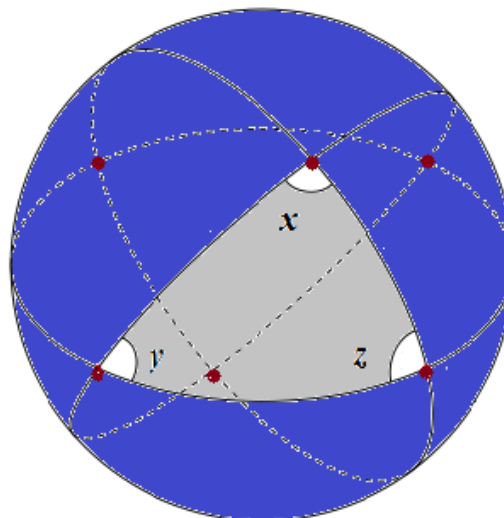
Elliptic geometry, also known as Riemannian geometry is a non-Euclidean geometry with positive curvature (because the surface swells out) and reinstates the parallel postulate of the Euclid's geometry with the statement [7]: Within a two-dimensional plane, for a given point P and a line l such that P does not lie on line l , then in the same plane there does not exist any line that is parallel to l passing through P . This implies that all lines in elliptic geometry intersect. Unlike the Euclidean or hyperbolic geometry, elliptic geometry is not a neutral geometry, which consists of Euclid's first four axioms. The architects of this geometry were Bernard Reimann and Ludwig Schläfli, who conceptualized the elliptic lines drawn like great circles that can be conceptualized as the longitudes of earth.



2.3 Figure with a positive curvature

There are certain unique properties of Elliptic geometry:

1. The sum of angles in a triangle is greater than 180° .
2. For any given two lines which are great circles, there are two intersecting points called *poles* or *antipodal points*. This notion was introduced by Felix Klein.



2.4 Circles representing lines in Elliptic geometry. Colored dots represent antipodal points.

3. A line has no end points and has finite length.
4. Any two triangles with same interior sum of angles have same area.

3 Hyperbolic Geometry Models

Hyperbolic models are used to define a hyperbolic plane in which all the axioms of hyperbolic geometry can be asserted. These models can be used to visualize hyperbolic diagrams in a Euclidean 2D space. There are three different models, discussed in the sections ahead, that are commonly used, namely: the Poincaré disk model, the Klein model and the Weierstrass model. The first two are the models that encapsulate finite hyperbolic geometry defined in a Euclidean 2D space, whereas the latter is an infinite model placed in a 3D Euclidean space. We can convert an item from one model to another model using a process called isomorphism explained ahead in [Section 3.4](#).

3.1 Poincaré Disk Model

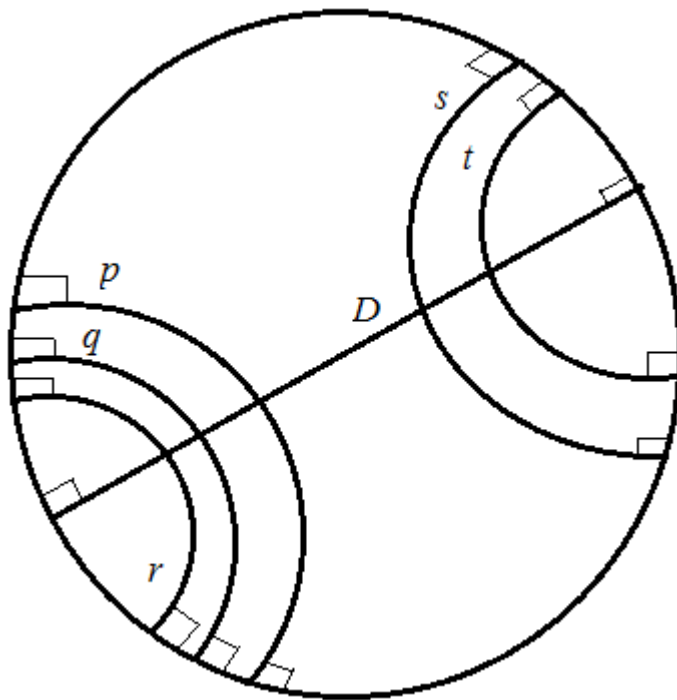
The Poincaré disk model represents conformal mapping of hyperbolic geometry, which means that this model preserves angles and does not distort them during transformations. The Poincaré disk model [8] was invented by Henry Poincaré, a French mathematician and philosopher. This model is also known as the conformal disk model. Below are a few salient properties of this model:

1. Any given point $P(x, y)$ in this model is equivalent to a Euclidean plane and lies within a Euclidean unit circle, which is a circle of radius one unit. Point $P(x, y)$ always satisfies the below condition:

$$x^2 + y^2 < 1$$

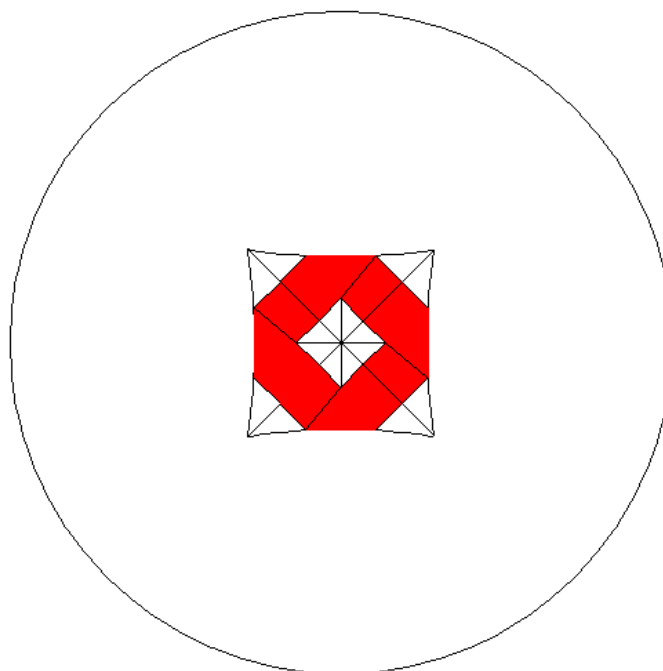
Euclidean points on the unit circle itself are also known as *ideal points*.

2. The lines in the model are drawn as arcs that meet the boundary orthogonally. Lines are categorized into two types: A chord or a segment passing through the centre of the Euclidean unit circle called a diameter, or an arc such that its ends points are perpendicular to the boundary of unit circle. Below is the diagram that shows arcs p , q , r , s and t and a diameter D , representing lines in this model.

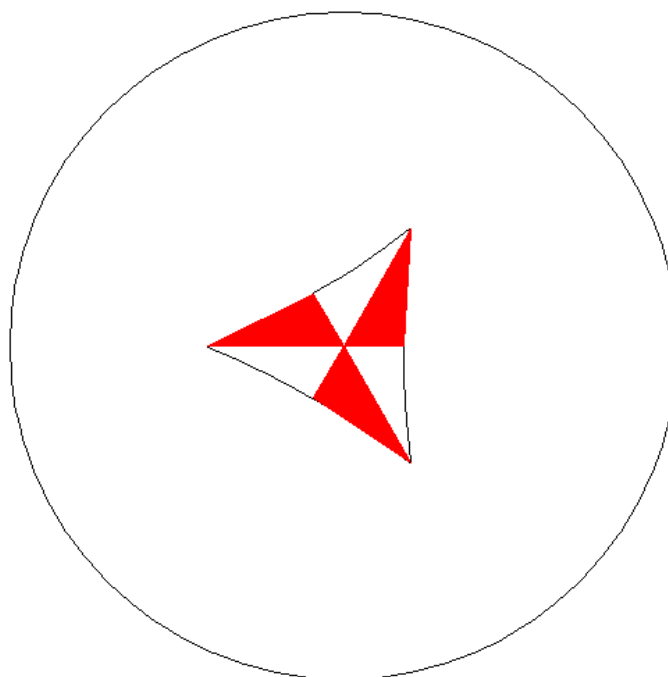


3.1 Lines in a Poincaré disk model

3. The terms "lie on" and "between" are similar in meaning to the Euclidean geometry.
4. Two intersecting lines can be parallel to the same line.
5. Below are examples of a few objects when viewed in this model:



3.2 A square in the Poincaré model (the red strips are part of a design pattern)



3.3 A triangle in the Poincaré model (the red strips are part of a design pattern)

3.2 Klein Model

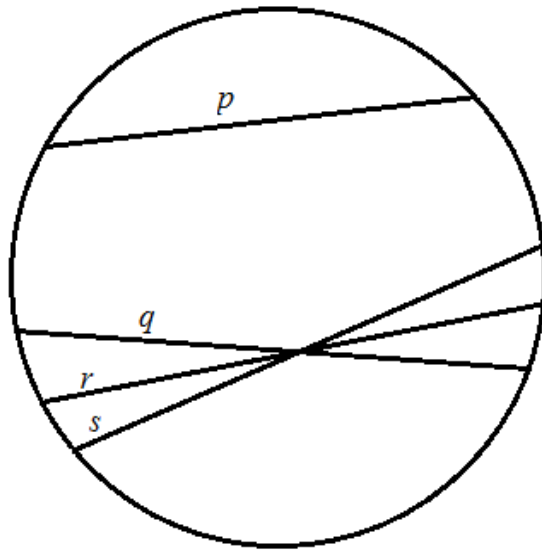
This model is a non-conformal disk model, meaning the angles are not accurately represented by this model, and the circles are distorted, as in a circular object in the Poincaré would be represented as an ellipse in this model. The Klein model is a projective disk model named after its inventor Felix Klein, a German mathematician. This model depicts a unit open disk in a Euclidean space where open chords correspond to hyperbolic lines. An open disc of radius r is a set of all points less than r from a fixed point in Euclidean space.

$D(x, r)$ is the mathematical symbol used to indicate an open disk. When $x=0$ (origin) and r is 1, it is considered to be an open unit disk. Below are a few important features of this model:

1. The points in the model are represented the interior of a unit disk, i.e., a point $P(x, y)$ satisfies the equations:

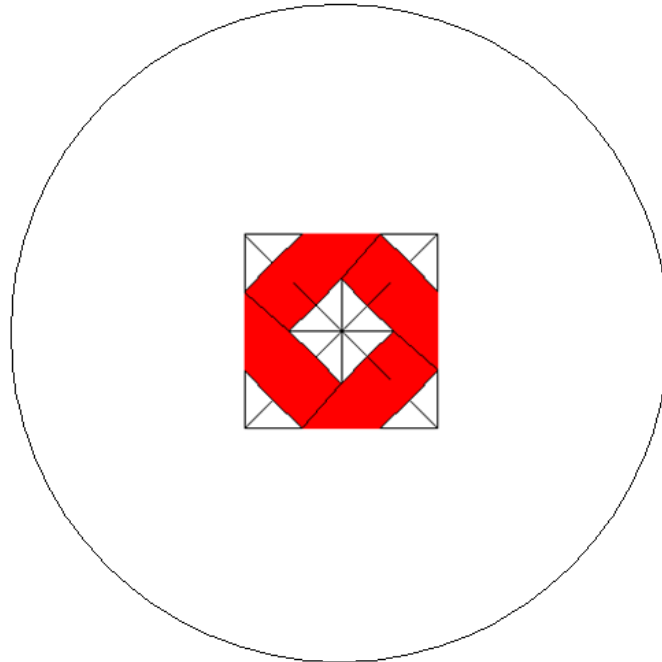
$$x^2 + y^2 < 1$$

2. The lines or segments are straight segments represented by open chords. An open chord [9] is defined as a closed chord having its end points removed. The figure shown below depict lines i.e., open chords $p, q, r,$ & s in Klein model:

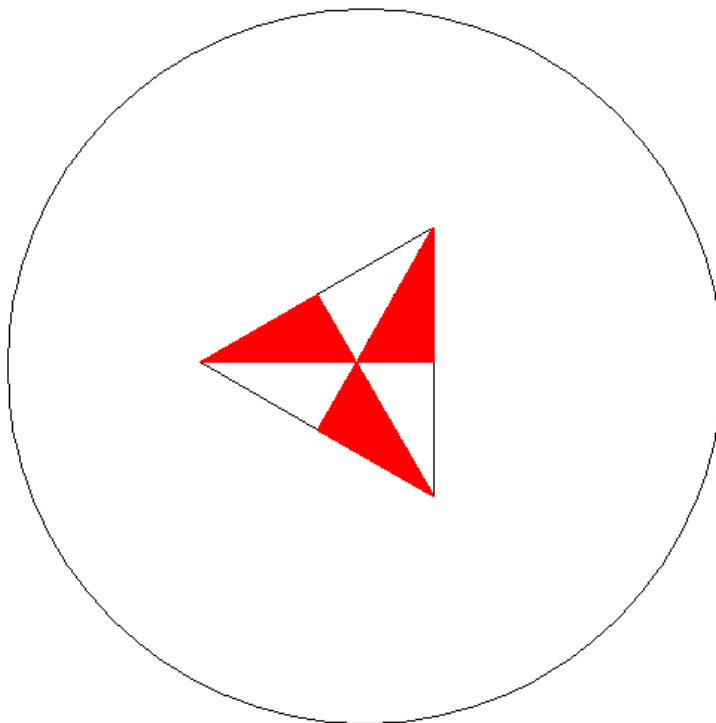


3.4 Lines(or open chords) in Klein model

3. Similar to the Poincaré model, the terms "lie on" and "between" are carry similar sense as in the Euclidean geometry.
4. Two lines are considered parallel if they fail to intersect.
5. Below are the figures that are counterparts to the figures 3.2 and 3.3



3.5 A square in Klein model



3.6 A triangle in the Klein model

3.3 Weierstrass Model

This model is used to represent the infinite model of hyperbolic geometry. The model is represented using a hyperbolic space projected onto the surface of a hyperboloid which can be extended up to infinite length in 3 dimensional space. This model can be easily projected, by making stereographic projections, onto the previous two models as it bears close relationship with them.

Given a vector $\mathbf{X}(x, y, z)$, the hyperboloid is this model can be represented mathematically as:

$$\langle \mathbf{X}, \mathbf{X} \rangle = x^2 + y^2 - z^2 = -k^2$$

Note: Given two vectors $\mathbf{X}(x_1, x_2, x_3)$ and $\mathbf{Y}(y_1, y_2, y_3)$, then $\langle \mathbf{X}, \mathbf{Y} \rangle$ is defined as:

$$\langle \mathbf{X}, \mathbf{Y} \rangle = x_1 y_1 + x_2 y_2 - x_3 y_3$$

This divides the hyperboloid into two sheets [10]: the upper sheet and lower sheet. The lower sheet is nothing but a reflection of the upper sheet, hence the lower sheet is not usually considered. Hence the points in the single sheet of the hyperboloid is represented as:

$$\langle \mathbf{X}, \mathbf{X} \rangle = -k^2 \text{ and } z > 0$$

A point in this model is represented as [11] the point that satisfies the equation for the upper sheet of the hyperboloid, which can also be written as:

$$x^2 + y^2 - z^2 = -1, \quad z > 1$$

Lines are the intersections of a Euclidean plane passing through the origin and the hyperboloid. A point L lies on the plane if it satisfies the below equation:

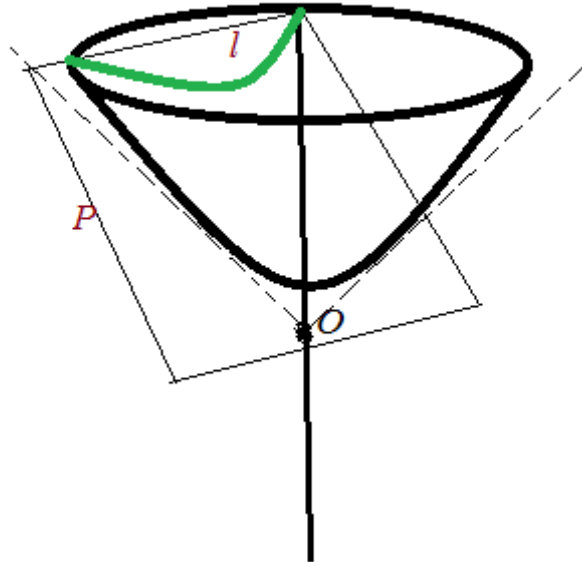
$$\langle \mathbf{X}, \mathbf{L} \rangle = 0 \text{ and } z > 0$$

A line l is generally represented by its pole using a 3 variable vector $[l_x \ l_y \ l_z]$ on

a hyperboloid satisfying the equation: $l_x^2 + l_y^2 - l_z^2 = +1$, then the line is a set of points satisfying the condition:

$$xl_x + yl_y - zl_z = 0$$

A straight line, also called as geodisc is obtained by the slicing the hyperboloid using a plane that passes through its origin. The diagram below shows an plane P forming a line l (a hyperbola indicated in green).



3.7 Every line in a Weierstrass model is a hyperbola

3.4 Isomorphisms

Isomorphism is a Greek word where in *iso* means equal and *morphis* means "to form". Isomorphism is the process of converting a object in model to its counterpart in another model by preserving the nature of relationship between the

elements in the objects i.e., by preserving its structure. The sections below explain the mechanism involved for different isomorphic relationships existing between the three aforementioned models.

3.4.1 Weierstrass & Poincaré Disk Model Isomorphism

A simple mapping exists [11] from the Weierstrass model to the Poincaré disk model, which is a unit disk in the XY plane. A stereographic projection is made from the Weierstrass model toward $(0, 0, -1)$, which is the vertex of the lower sheet of the hyperboloid. The formula is given as follows:

$$(x, y, z) \rightarrow (x/(1 + z), y/(1 + z), 0)$$

The inverse mapping from Poincaré to Weierstrass is given as follows:

$$(x, y) \rightarrow (2x/(1 - s), 2y/(1 - s), (1 + s)/(1 - s)), \text{ where } s = x^2 + y^2$$

In this project, for drawing the hyperbolic central polygons, all the calculations are made in the Weierstrass model whereas the output is shown in the Poincaré model.

3.4.2 Weierstrass & Klein Model Isomorphism

Objects from the Weierstrass model can be mapped onto the Klein model by stereographic projection of the Weierstrass model onto the $Z = 1$ plane towards the point $(0, 0, 0)$ using the below formula:

$$(x, y, z) \rightarrow \left(\frac{x}{z}, \frac{y}{z}, 1\right)$$

Conversely, a point $(x, y, 0)$ in Klein model can be mapped to Weierstrass model

using the formula:

$$(x, y, 0) \rightarrow \frac{1}{(1 - x^2 - y^2)}(2x, 2y, 1 + x^2 + y^2)$$

3.4.3 Poincaré Disk & Klein Model Isomorphism

Both of these models are used to represent hyperbolic objects within a unit circle. Poincaré disk model represents a hyperbolic line using hyperbolic arcs orthogonal to the boundary of the circle, whereas the Klein model represents the same object using Euclidean line segments which are also called open chords.

A vector U in Poincaré disk model, can be converted to its corresponding vector V in Klein model using the below equation:

$$V = \frac{2U}{1 + U \cdot U}$$

Similarly, a vector V in Klein model can be converted to its corresponding vector U in Poincaré model using the equation:

$$U = \frac{V}{1 + \sqrt{1 - V \cdot V}}$$

The above formula can also be written as:

$$U = \frac{V(1 - \sqrt{1 - V \cdot V})}{V \cdot V}$$

This is an important isomorphic relationship with respect to this research focus, because the Euclidean counterparts of the hyperbolic images are generated by first

converting the points in the Poincaré disk model to the Klein model. Also, the arcs are converted to Euclidean line segments.

4 Euclidean Patterns

In this section we explain the terminology involved in the patterns we desire to create. First, we explain the basic process of tessellation, and second, we explain the idea of fundamental region or motif, that forms the crux of the pattern creation. Lastly we discuss and define wallpaper groups. In the next chapter we discuss the methodology involved in creating the central Euclidean polygon and the subsequent step of tiling it.

4.1 Tessellation

A tessellation or tiling is formed when a shape is repeatedly redrawn until the entire plane is filled leaving no gaps or holes. The shape could be translated into congruent pieces and repeated in a mosaic or checkered pattern. Tessellation is derived from a Greek word "tesseres" which means "four" in English. This is because the very first patterns were made of squares. A regular tessellation is indicated by the Schläfli symbol $\{p, q\}$, which means a tiling of a plane is done by regular p -sided polygons meeting edge to edge and vertex to vertex, with q meeting at a vertex.

If $(p-2)(q-2) > 4$, the tessellation is hyperbolic. The only Euclidean tessellations are $\{3, 6\}$, the tessellation by equilateral triangles, $\{4, 4\}$, the tessellation by squares, and $\{6, 3\}$, the tessellation by regular hexagons. The goal of this research is to transform hyperbolic patterns based on $\{p, q\}$ tessellation to repeating Euclidean patterns where possible. This can be done in three cases. Hyperbolic patterns based on $\{3, q\}$, $\{4, q\}$ and $\{6, q\}$ can be transformed to $\{3, 6\}$, $\{4, 4\}$ and $\{6, 3\}$ patterns respectively.

4.2 Fundamental Region and Motif

A *motif* is the smallest shape that is used to generate the center polygon. In fact, an entire repeating pattern can be created using congruent copies of fundamental pattern, also known as the *motif*. Reflection and rotation of the motif creates the central polygon, also called Fundamental region. We use the term *p-gon* to indicate a regular p-sided hyperbolic polygon.

4.3 Wallpaper Group

A wallpaper group is created when a tessellation is made with congruent regular polygons on the Euclidean plane. The term "*regular*" means all the sides and the angles in the polygon are same and the term "*congruent*" means the copies of polygons used to create the tiling are all of same size and shape. A wallpaper group, also known as a plane crystallographic group, is a type of discrete group of isometries of the Euclidean plane. There are four kinds of isometries [14]:

1. Translations, where in Euclidean plane is moved by a certain distance.
2. Rotation, where in the plane is rotated by a certain angle of rotation about a point.
3. Reflection, obtained by reflecting the plane about a particular axis.
4. Glide reflections, is a combination of reflection and translation.

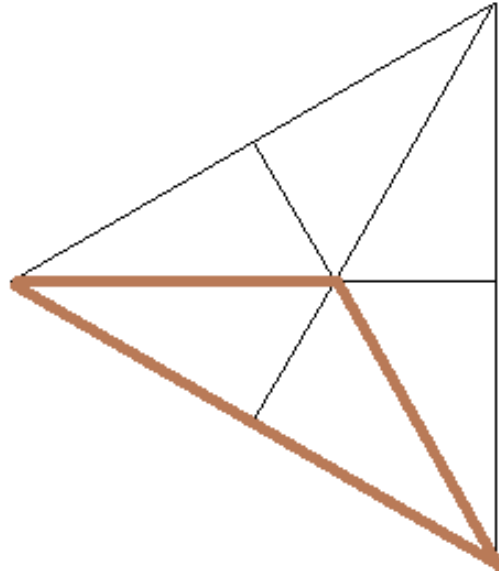
For this project, we focus our work on creating tilings for equilateral triangles, squares, and regular hexagons.

5 Implementation

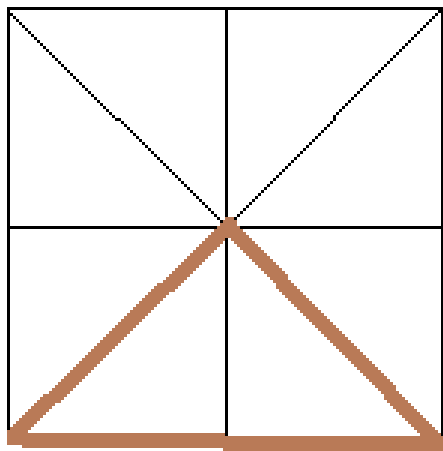
This chapter involves discussion and explanation about the methodology involved in creating the central Euclidean polygon. We first explain the process of mapping objects from hyperbolic space to Euclidean space and then explain the algorithm, proposed by Dr. Dunham, to create the central Euclidean pattern. Subsequently, we describe the logic used in the creation of tiling the Euclidean patterns.

5.1 Transforming Poincaré Model Points To Klein Model

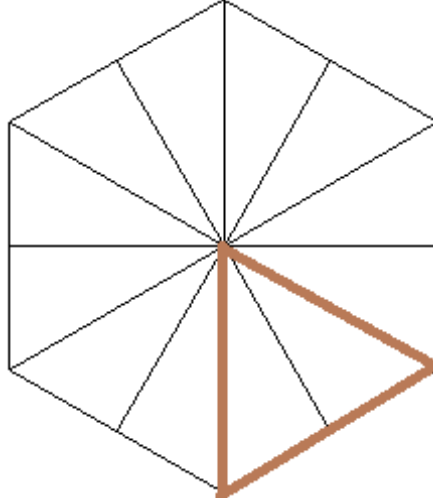
The Poincaré points are mapped to the Klein model using the isomorphism technique explained in [Section 3.5.3](#). The central polygon is divided into p isosceles triangles and they are drawn using Euclidean lines which forms the framework for the pattern. These Euclidean lines are approximations to the hyperbolic arcs in the Poincaré model. The images below show the three kinds of Euclidean polygons created for different triangles, squares and hexagons. These form the root for the actual pattern that is created, which is explained in the next section.



5.1 The central Euclidean triangle divided into three isosceles triangles of similar size (indicated in brown)



5.2 The central Euclidean square divided into four isosceles triangles of similar size (indicated in brown)



5.3 The central Euclidean hexagon divided into six equilateral triangles of similar size (indicated in brown)

Below is the logic involved in the creation of each of the above Euclidean polygons:

1. We iterate through the total number of sides of the polygon, .i.e., p . For every iteration i , starting from 1 until p , a total of 3 Euclidean polylines are drawn. Therefore a total of $3 \times p$ total lines drawn. This is because each subdivided isosceles triangle is composed of three lines segments. (refer to the example that immediately follows this algorithm)
2. For every iteration j starting from 0 until 3, the x-coordinate and y-coordinate (x, y) of the Euclidean lines are calculated as follows:

$$x_j = \cos \theta_i \times X_j - \sin \theta_i \times Y_j$$

$$y_j = \sin \theta_i \times X_j + \cos \theta_i \times Y_j$$

where $\cos \theta_i$ and $\sin \theta_i$ for every i^{th} iteration is calculated as follows:

$$\cos \theta_i = \cos \left(\frac{2 \times i \times \pi}{p} \right)$$

$$\sin \theta_i = \sin \left(\frac{2 \times i \times \pi}{p} \right)$$

3. Two one dimensional matrices of size 4, for storing values of X_j and Y_j , for $j = 0$ to 3, are initially created. They are initialized as follows:

$$\text{Values for } X_{0 \text{ to } 3}: \quad [xqpt \quad xqpt \quad 0 \quad x2pt]$$

$$\text{Values for } Y_{0 \text{ to } 3}: \quad [yqpt \quad -yqpt \quad 0 \quad 0]$$

4. The sequential calculations involved for $xqpt$, $yqpt$ and $x2pt$ are:

$$\cos \theta = \cos \left(\frac{\pi}{p} \right)$$

$$\sin \theta = \sin \left(\frac{\pi}{p} \right)$$

$$\cos 2\theta = \cos \left(\frac{2\pi}{p} \right)$$

$$\sin 2\theta = \sin \left(\frac{2\pi}{p} \right)$$

$$\cosh q = \frac{\cos \left(\frac{\pi}{q} \right)}{\sin \theta}$$

$$\sinh q = \sqrt{(\cosh q \times \cosh q) - 1}$$

$$\cosh 2 = \frac{1}{\left(\frac{\sin \left(\frac{\pi}{p} \right)}{\cos \left(\frac{\pi}{p} \right)} \right) \left(\frac{\sin \left(\frac{\pi}{q} \right)}{\cos \left(\frac{\pi}{q} \right)} \right)}$$

28

$$\sin h2 = \sqrt{(\cosh2 \times \cosh2) - 1}$$

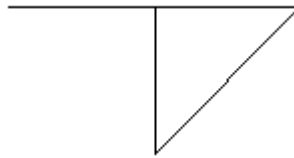
$$radius2 = \frac{\sinh2}{1 + \cosh2}$$

$$x2pt = \frac{\sinh q}{1 + \cosh q}$$

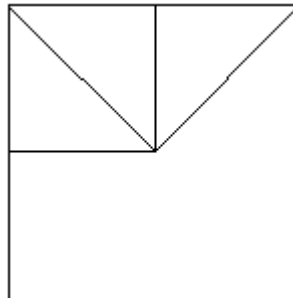
$$xqpt = \cos \theta \times radius2$$

$$yqpt = \sin \theta \times radius2$$

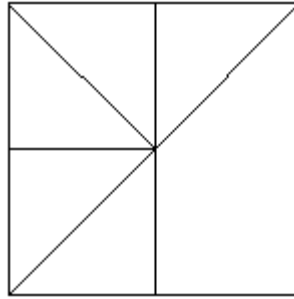
For the [Figure 5.2](#), where $p = 4$, below are the sequence of diagrams created for every i^{th} iteration from 1 to 4:



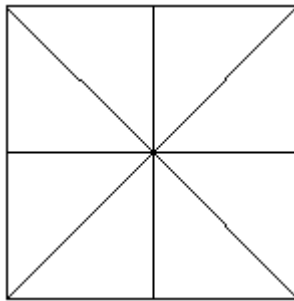
5.4 Partially completed central Euclidean square, composed of three line segments, after the 1st iteration. This indicates the starting figure.



5.5 Partially completed central Euclidean square after the 2nd iteration



5.6 Partially completed central Euclidean square after the 3rd iteration



5.7 Fully completed central Euclidean square after the final(4th) iteration

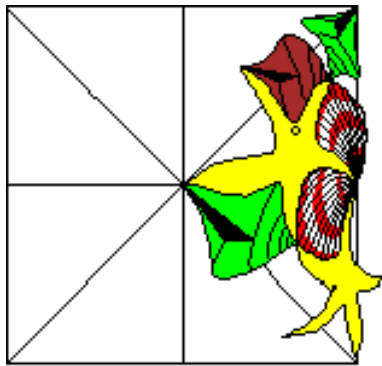
Once the Euclidean polygon is drawn, the pattern for the object is drawn using the motif points by reflecting and rotating the motif. The algorithm is explained in the following sub-section.

5.2 Generation Of Central Euclidean Pattern Algorithm

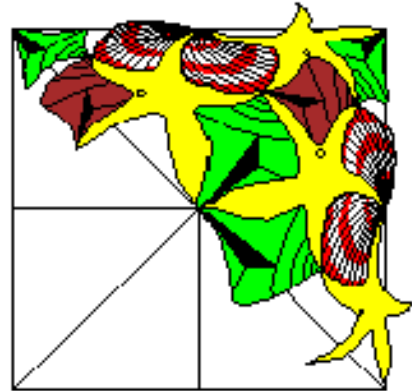
Different copies of the fundamental pattern are made based on the transformations involved, upon the framework created in previous step which acts as a scaffolding.

Transformations are of two types [12]: Reflection and rotation about the origin :

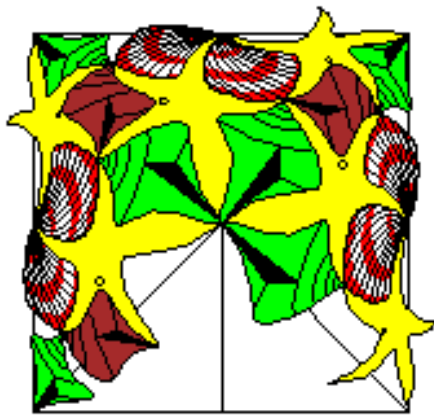
the p-gon center. The reflections are sub-divided into two types: Reflection across the p-gon radius (or the hypotenuse) and reflection across the perpendicular bisector of a p-gon edge. If there is no reflection symmetry, the fundamental pattern is rotated about the origin to fill the fundamental region. If the reflection symmetry exists, then the fundamental pattern is first reflected based on the symmetry and then the copies of this new pattern is rotated about the origin to create the central polygon. The diagrams below depict each of the three cases for creation of a central polygon:



(a)



(b)

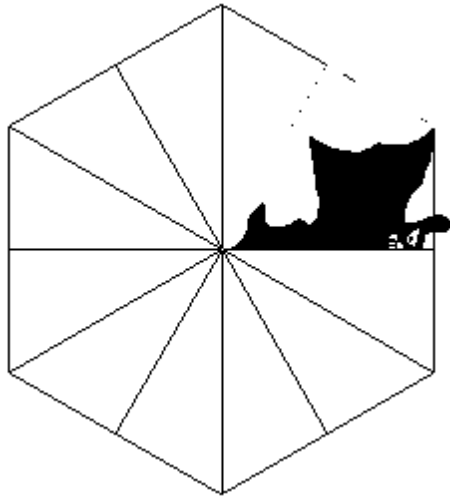


(c)

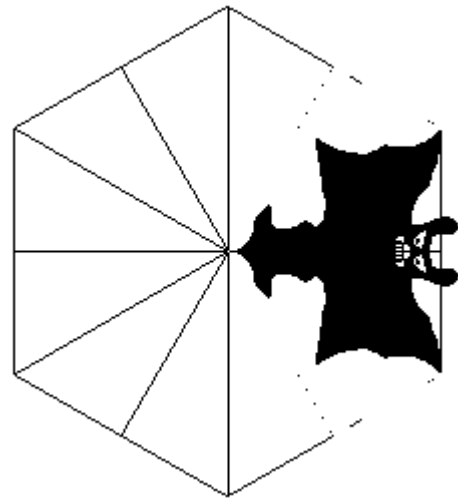


(d)

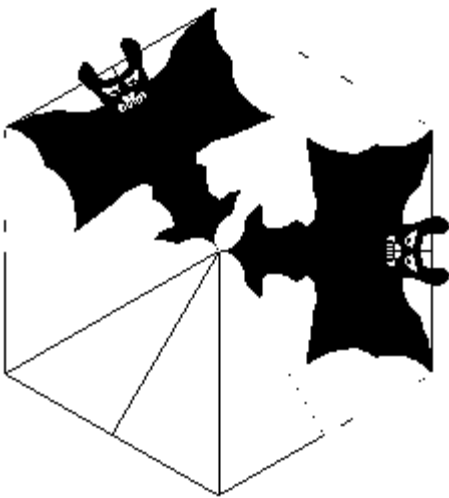
5.8 Rotation (only) of motif about the origin starting from step (a) until (d) to create the central polygon.



(a)



(b)

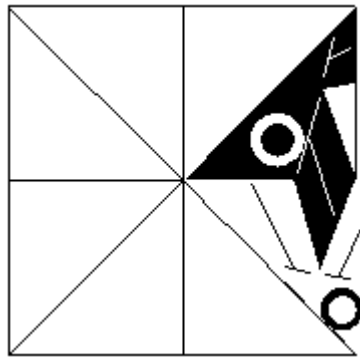


(c)

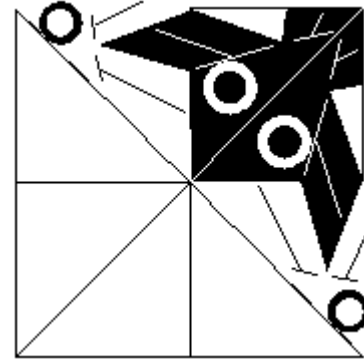


(d)

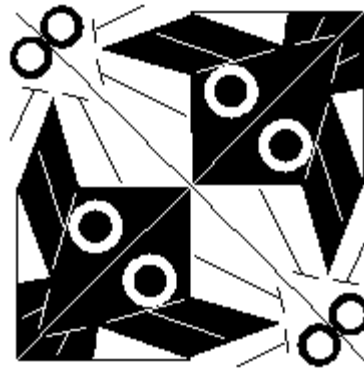
5.9 Reflection across the edge bisector. (a) Initial motif created.
 (b) Reflection of actual motif created in step a across the edge.
 (c) Motif created in step b is rotated about the origin.
 (d) Motif created in step b is rotated about the origin once again to complete the central polygon.



(a)



(b)



(c)

5.10 Reflection across a hypotenuse. (a) Initial motif created
 (b) Reflection of motif created in step a about its hypotenuse.
 (c) Final polygon created by rotating motif created in step (b).

Next we describe the implementation of the logic involved in rotating and reflecting the points. We discuss the structures of different matrices that assist in creating the copies of reflected and rotated motifs.

Based on the number of sides next to the motif in the central region (referred to

as s in the sample data file format given in [Appendix A](#)) and the value of p from the input data file, the total number of rotations to be performed is calculated as:

$$numOfRotations = \frac{p}{s}$$

The points are transformed using the following algorithm:

1. If the p -gon pattern is reflected across an edge bisector, go to step 2a., or if it can be reflected using its hypotenuse, then go to step to 2b, otherwise go to the step 3.
2. a. Reflect the motif by obtaining the new points using the following matrix multiplications:

$$[x_{new} \ y_{new}] = [x_{actual} \ y_{actual}] \times \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- b. In this case, the new points are reflected using the below matrix multiplications :

$$[x_{new} \ y_{new}] = [x_{actual} \ y_{actual}] \times \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$$

Calculations for $\cos 2\theta$ and $\sin 2\theta$ values are shown in [section 5.1](#).

3. Define *rotationMatrix* as $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
4. For every iteration starting from $i = 1$:
 - i. If i is less than or equal to $numOfRotations$, then to step ii, otherwise exit the loop.
 - ii. Generate the motif about the origin by multiplying it with the *rotationMatrix* :

$$[x_{motif} \ y_{motif}] = [x_{motif} \ y_{motif}] \times rotationMatrix$$

iii. Re-define the *rotationMatrix* using the below calculations :

$$rotationMatrix = rotationMatrix \times \begin{bmatrix} \cos\left(\frac{2\pi s}{p}\right) & -\sin\left(\frac{2\pi s}{p}\right) \\ \sin\left(\frac{2\pi s}{p}\right) & \cos\left(\frac{2\pi s}{p}\right) \end{bmatrix}$$

where s is the number of sides in the central region described earlier.

iv. Increment i by 1 i.e., i becomes $i + 1$, and go to step i.

Once this algorithm is executed, we move to the tiling aspect of the process.

5.3 Tiling Euclidean polygons

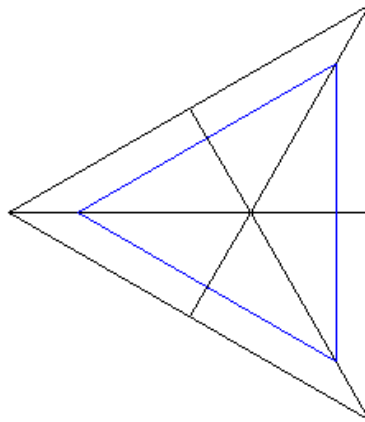
Once the Euclidean pattern is drawn, we then tessellate it across the Euclidean plane to create different tiling patterns. In this section, we explain the process of creating such tilings starting off with equilateral triangles and then squares, followed by regular hexagons.

5.3.1 Tiling Euclidean Polygon Of Type: Equilateral Triangle

Tiling a triangular Euclidean pattern (based on $\{3, 6\}$) is a three step process that primarily involves creation of copies of Euclidean pattern (using the process described in the above two sections), however the x-coordinate and the y-coordinate of every copy of the pattern is moved by a certain distance parameter.

The first step in this process is to rotate the original Euclidean pattern in clockwise direction by 90° to obtain the Euclidean pattern we wish to tile. For instance,

consider the triangle below:



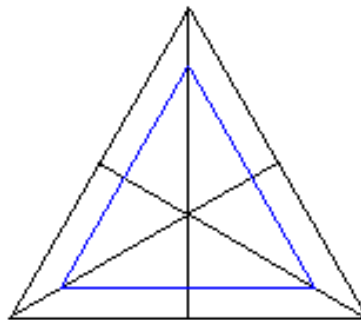
5.11 {3, 6} Euclidean triangle

This triangle is rotated by modifying the original *x-coordinate* and the *y-coordinate* values of the motif as :

$$\text{new } x \text{ coordinate} = \text{actual } y \text{ coordinate}$$

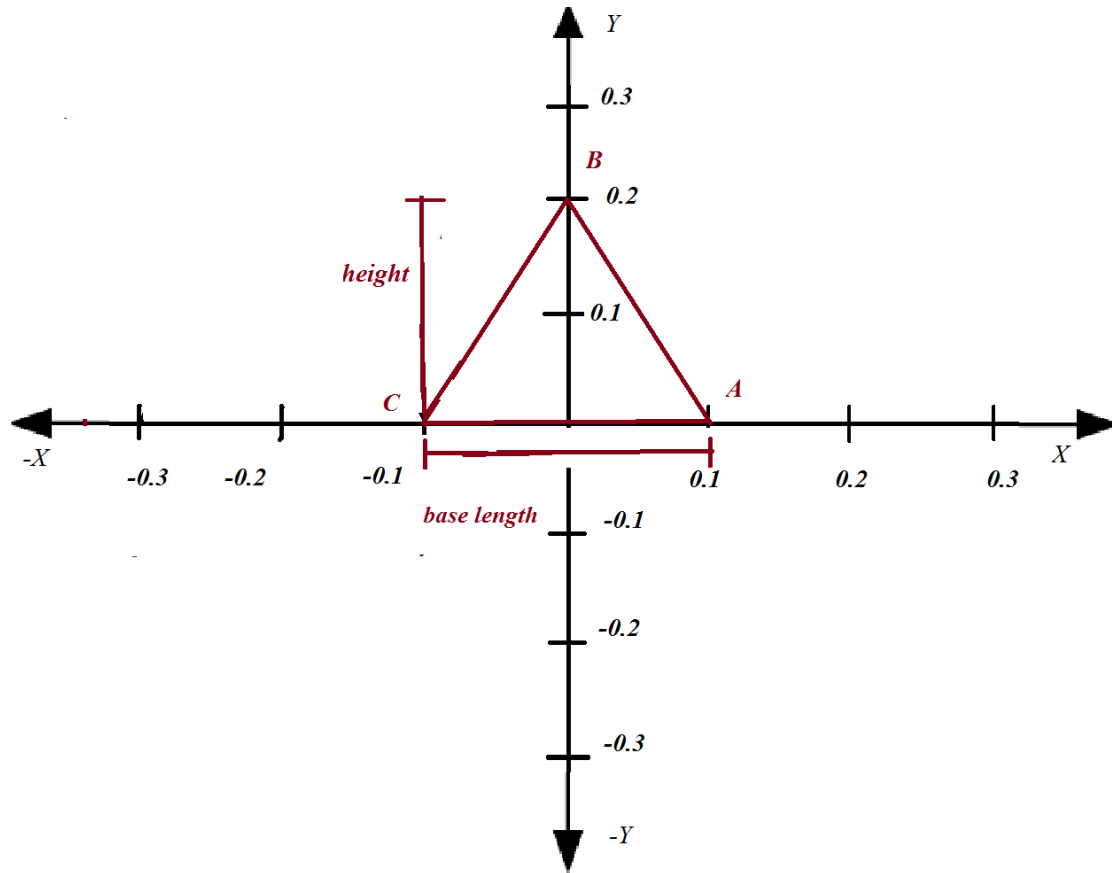
$$\text{new } y \text{ coordinate} = -(\text{actual } x \text{ coordinate})$$

This would generate the below image:



5.12 Rotated {3, 6} Euclidean triangle

Based on the logic explained in [Section 5.1](#), let's say a triangle with points A, B & C is drawn on an XY plane as shown in the image below:



5.13 Euclidean triangle on an XY plane

The length of the triangle's base can be calculated using the x-coordinates of the points A and C as:

$$\mathbf{baseLength} = x\ coordinate_A - x\ coordinate_C$$

The **height** of the triangle is the y-coordinate of point B. Based on these distances the algorithm below is used to create a tiling across the plane:

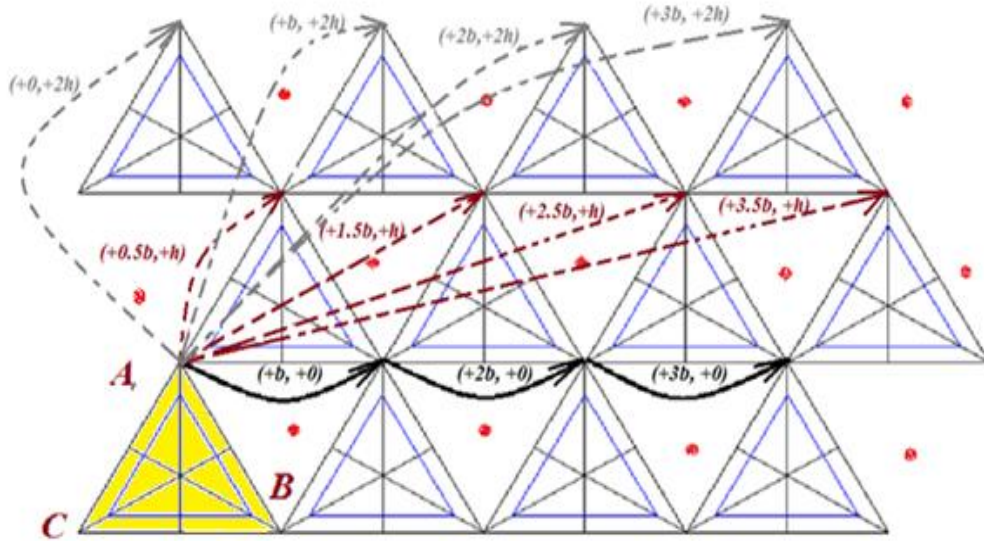
1. Initialize parameters $loop_v = 0, loop_h = 0$
2. If $loop_v$ is less than or equal to 2, then re-assign $loop_h = 0$ and go to step 3, otherwise exit out of the algorithm.
3. Assign $verticalDistanceToMove$ to $(height \times loop_v)$

4. If $loop_h$ is less than or equal to 3, then go to step 5, otherwise, go to step 8.
5. If $loop_v$ is 0 or 2, go to step 5a, other go to step 5b.
6. a. Assign $horizontalDistanceToMove$ to $(baseLength \times loop_h)$
b. Assign $horizontalDistanceToMove$ to $(baseLength \times (loop_h + 1))$.
The is because in the second layer we need to align the vertices.
7. Redraw the Euclidean pattern by moving the original (x, y) values of the motif points to (x_{move}, y_{move}) as follows:
$$x_{move} = x + horizontalDistanceToMove$$

$$y_{move} = y + verticalDistanceToMove$$
8. Increment the value of $loop_h$ by 1, then go to step 4.
9. Increment the value of $loop_v$ by 1, then go to step 2.

The values $loop_v$ and $loop_h$ are used to keep a track of the outer and the inner loops respectively. The inner loop is for shifting the x-coordinate value, while the outer loop affects the y-coordinate.

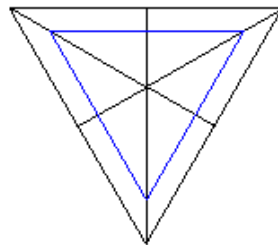
Below is the intermediate tiling pattern created after completing the above algorithm:



5.14 Detailed intermediate result in tiling a $\{3, 6\}$ triangle using the above algorithm.

The colored arrows indicate the fashion in which the (x, y) coordinates of the original Euclidean triangle ABC(highlighted in yellow), are incremented. The red dots indicate the gaps or holes in the pattern which are yet to be filled.

The second step involved in this process is to fill in the gaps created. The original points are modified such that an inverse of the original pattern is created about its base, as shown below:



5.15 Inverse of the Euclidean pattern of shown in figure 5.10

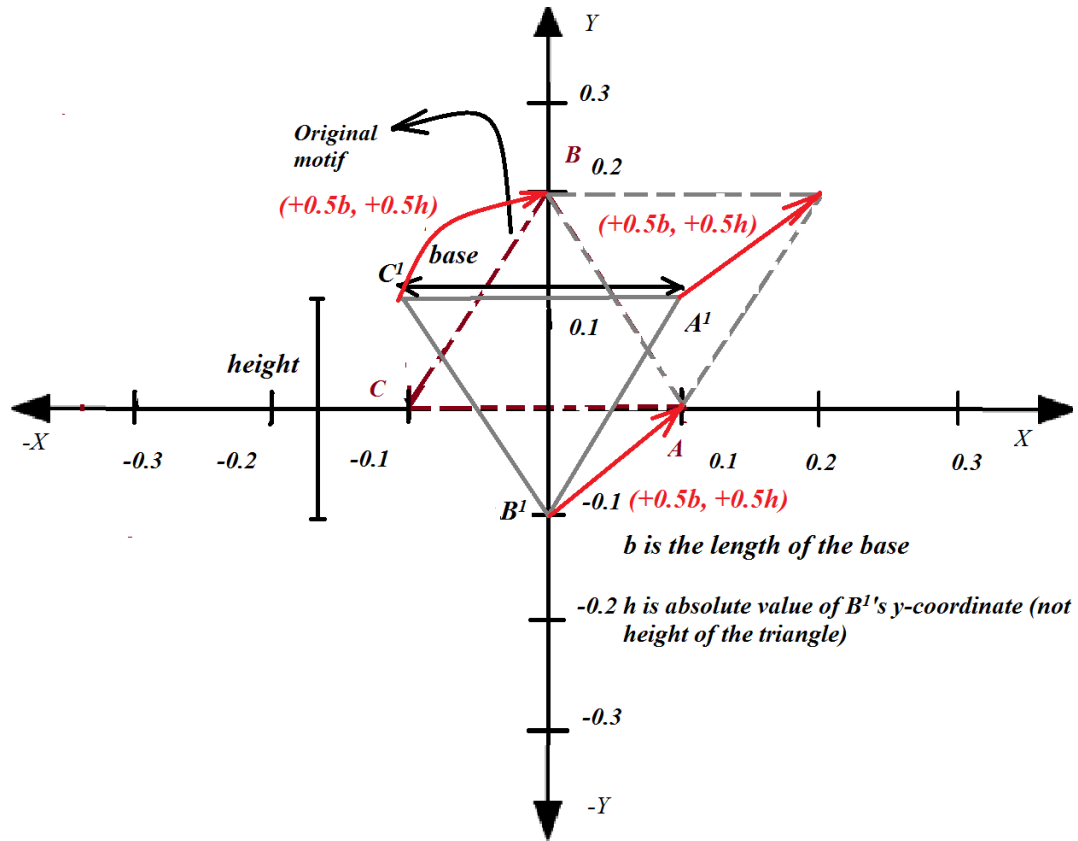
This is obtained by modifying the original x -coordinate and the y -coordinate

values of the motif as :

$$\text{new } x \text{ coordinate} = -(\text{actual } y \text{ coordinate})$$

$$\text{new } y \text{ coordinate} = \text{actual } x \text{ coordinate}$$

The algorithm to create inverse patterns and fill the gaps is slightly different compared to the previous one, because, the length of *height* and *base* of the triangle that we start from are different. For instance, below is the inverse of the Euclidean pattern in XY grid



5.16 Detailed image of the inverse of the Euclidean triangle in XY plane

The dotted maroon triangle depicts the actual Euclidean pattern whereas the solid grey triangle represents its inverse that we have generated by changing its (x, y) coordinates. The inverse tiling should start from the position indicated by the dotted grey

triangle. The $(+b, +h)$ values indicated are the distances to be added to the actual (x, y) values to level up the pattern, where b is the length of the base of the triangle and h is the absolute value of point B's y co-ordinate.

The length of the triangle's base can be calculated using the x-coordinates of the points A and C as:

$$\mathbf{baseLength} = x\ coordinate_A - x\ coordinate_C$$

Define ***height*** as the absolute value *y-coordinate* of point B. Using these distances below is the algorithm used to create inverse tiling across the plane:

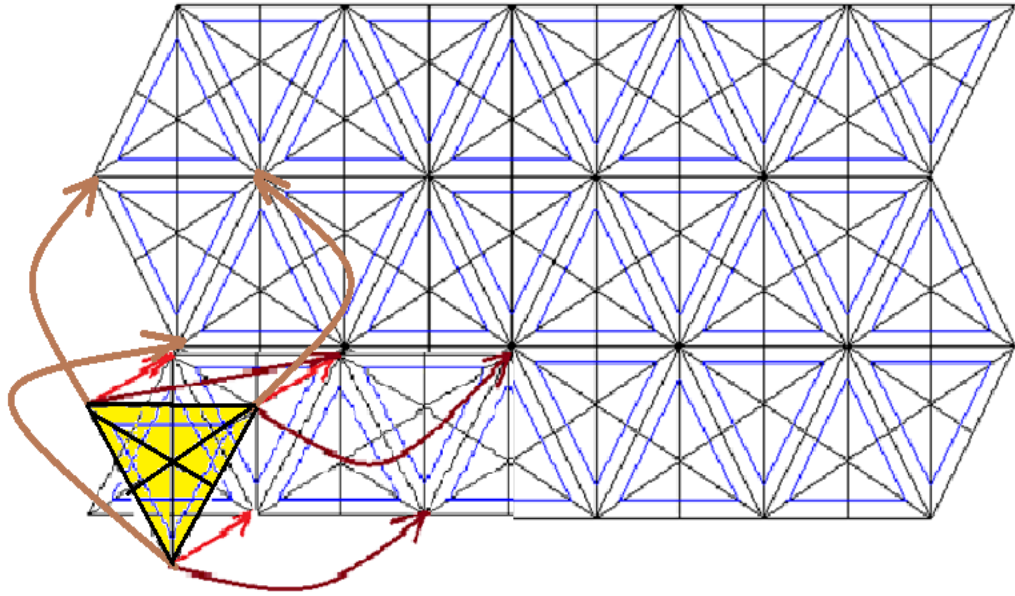
1. Initialize parameters $loop_v = 0, loop_h = 0$
2. If $loop_v$ is less than or equal to 2, then re-assign $loop_h = 0$ and go to step 3, otherwise exit out of the algorithm.
3. Assign *verticalDistanceToMove* to $((3 \times loop_v) + 1) \times \mathbf{height}$
4. If $loop_h$ is less than or equal to 3, then go to step 5, otherwise, go to step 8.
5. If $loop_v$ is 0 or 2, go to step 5a, otherwise go to step 5b.
6. a. Assign *horizontalDistanceToMove* to $(baseLength \times ((2 \times loop_h + 1)$
- b. Assign *horizontalDistanceToMove* to $(baseLength \times (loop_h + 1) \times 2)$. This is done because in the second layer we need to align the vertices.
7. Redraw the Euclidean pattern by moving the original (x, y) values of the motif points to (x_{move}, y_{move}) as follows:

$$x_{move} = x + horizontalDistanceToMove$$

$$y_{move} = y + verticalDistanceToMove$$

8. Increment the value of $loop_h$ by 1, then go to step 4.
9. Increment the value of $loop_v$ by 1, then go to step 2.

Similar to the previous algorithm described the values $loop_v$ and $loop_h$ are used to keep a track of the outer and the inner loops respectively. The inner loop creates inverse copies of pattern horizontally, while the outer loop helps in shifting the pattern upward. Consider the below image:



5.17 Inverse tiling of $\{3, 6\}$ Euclidean triangle

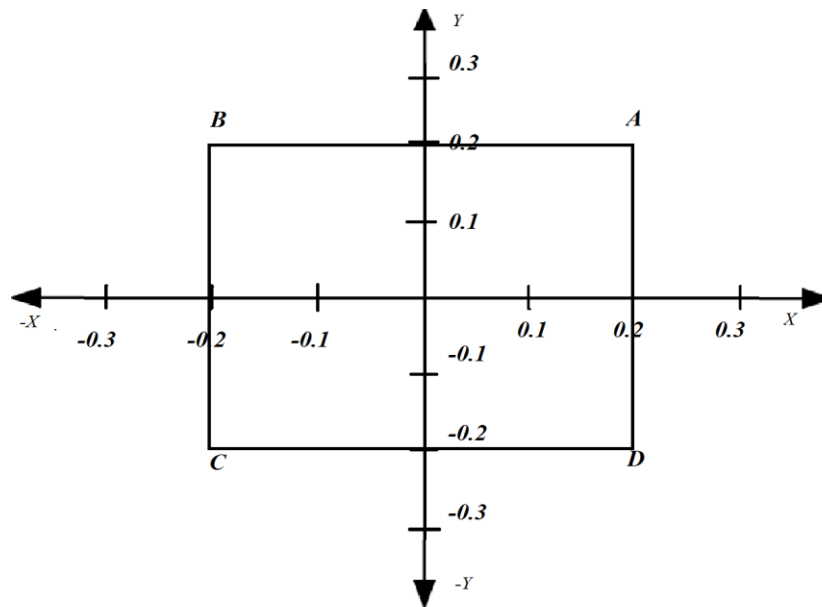
The highlighted pattern in yellow with black lines, in the bottom-most row is the actual position from which the pattern is shifted after every iteration. The red arrows indicate the triangle's position after the first iteration, and the maroon arrows indicate the position after the second iteration. In a similar fashion, patterns in the second and third row are completed. However in the second row, the inverse pattern is created starting

from the left of the actual pattern, as indicated by a brown arrow.

5.3.2 Tiling Euclidean Polygon Of Type: Square

The algorithm involved in the tiling Euclidean square patterns (i.e., based on $\{4, 4\}$) is very much equivalent that of triangles. We do not have to work with the overhead of creating the inverse copies of pattern, because the original square is drawn, the subsequent copies are made in using the length of the side and moving the original points by adding values in an arithmetic progression as described below:

Consider the Euclidean square in an XY plane:



5.18 Euclidean square in XY plane.

The length of the size can be calculated as :

$$sideLength = \text{Distance between point A and Point B}$$

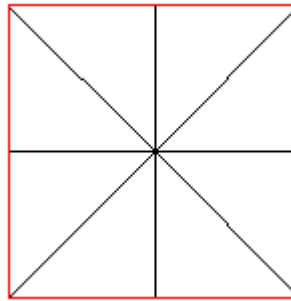
The tiling algorithm is then implemented as follows:

1. Initialize parameters $loop_v = 0$, $loop_h = 0$

2. If $loop_v$ is less than or equal to 2, then re-assign $loop_h = 0$ and go to step 3, otherwise exit out of the algorithm.
3. Assign $verticalDistanceToMove$ to $(sideLength \times loop_v)$
4. If $loop_h$ is less than or equal to 3, then go to step 5, otherwise, go to step 8.
5. Assign $horizontalDistanceToMove$ to $(sideLength \times loop_h)$
6. Redraw the Euclidean pattern by moving the original (x, y) values of the motif points to (x_{move}, y_{move}) as follows:
$$x_{move} = x + horizontalDistanceToMove$$

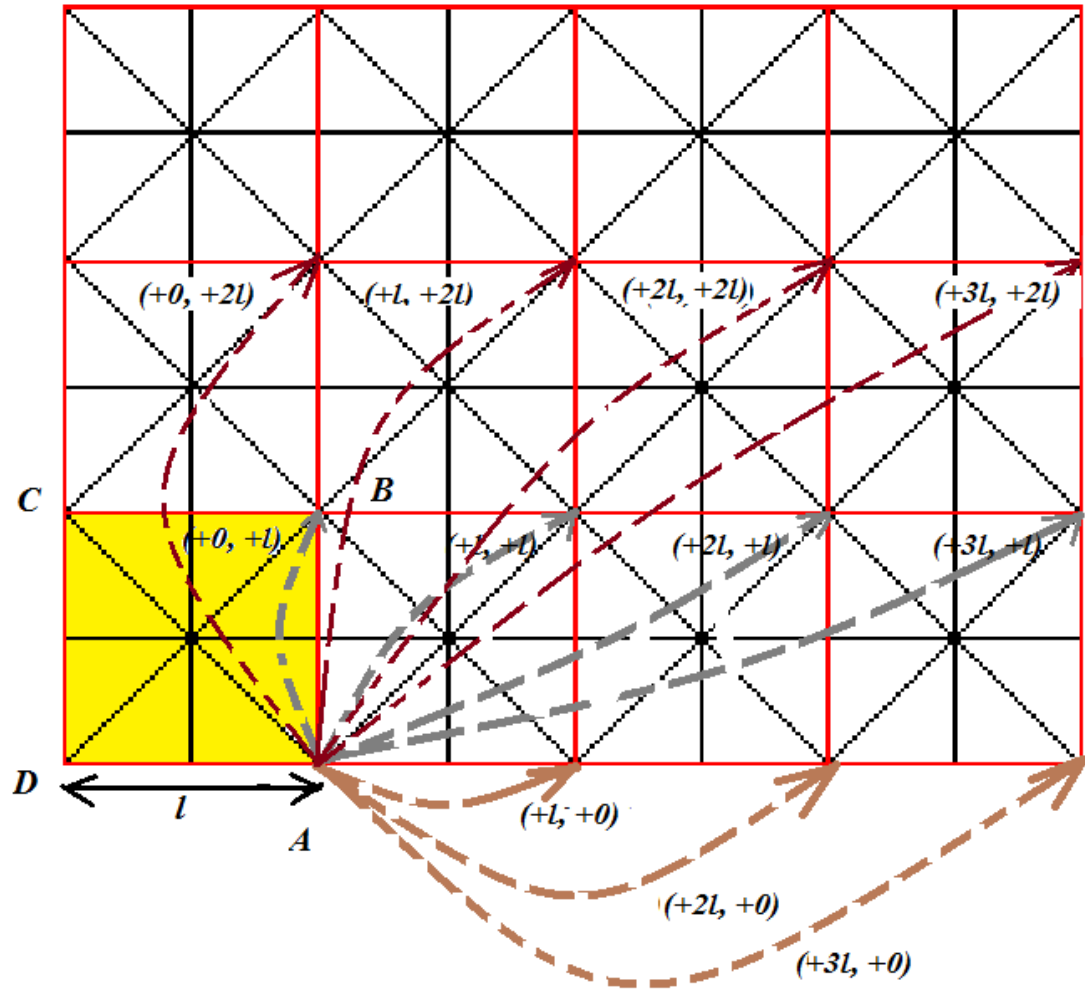
$$y_{move} = y + verticalDistanceToMove$$
7. Increment the value of $loop_h$ by 1, then go to step 4.
8. Increment the value of $loop_v$ by 1, then go to step 2.

Consider the Euclidean square below:



5.19 A $\{4, 4\}$ Euclidean square

After tiling the above pattern, we obtain the image below:

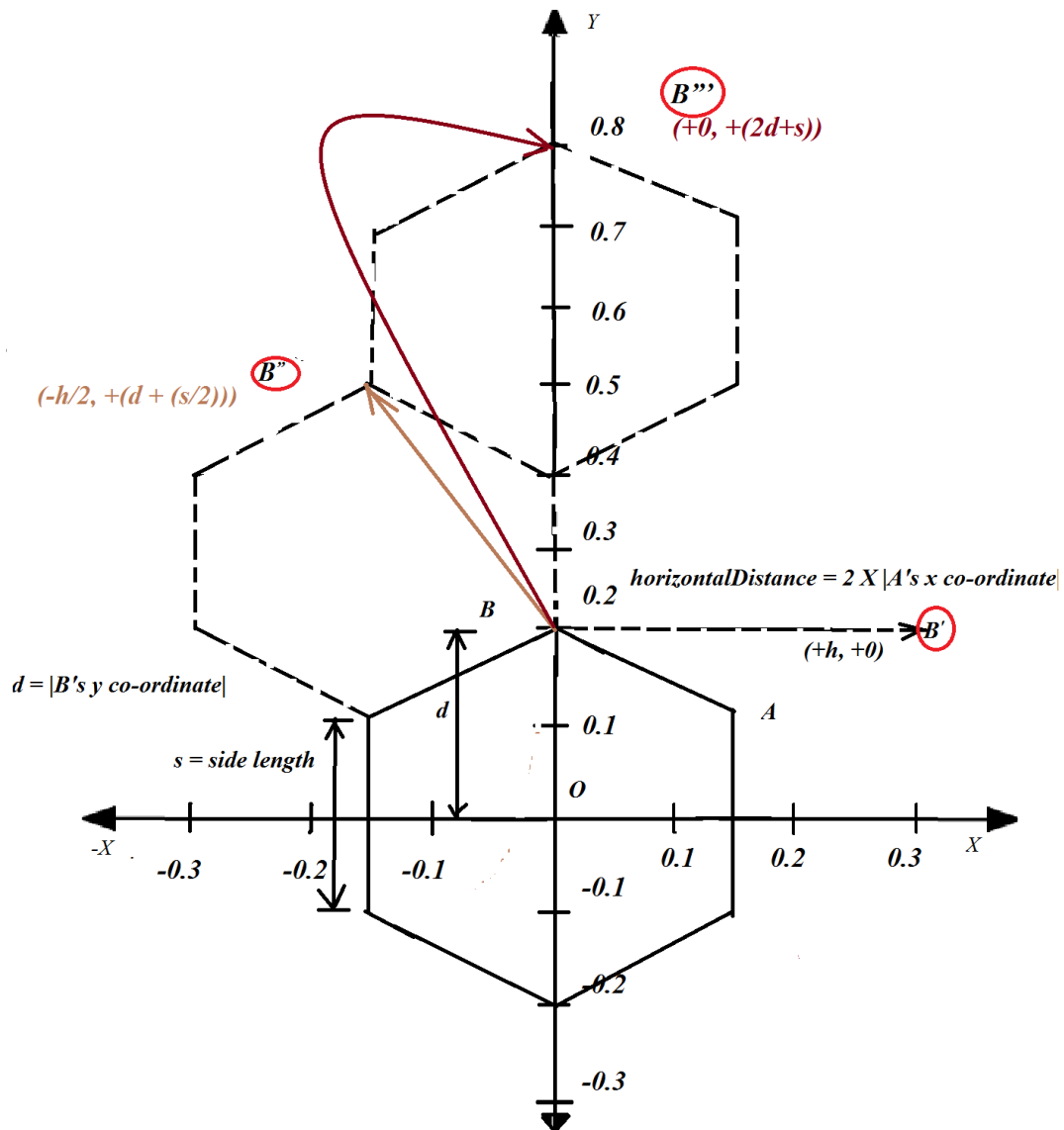


5.20 Detailed Tiling of $\{4, 5\}$ Euclidean square

The values mentioned on the dotted arrows indicate the distances to be added to the (x, y) coordinates of the motif points to move the pattern into particular position the arrow is pointing towards, and l is length of the square's side. The position of the original pattern in its actual position is highlighted in yellow.

5.3.3 Tiling Euclidean Polygon Of Type: Regular Hexagon

The mechanism involved in tiling Euclidean patterns of type regular hexagon (based on $\{6, 3\}$) is quite similar to the previous mechanisms described for squares and triangles with slight variations. Consider the below Euclidean hexagon on an XY grid:



5.21 Detailed hexagon on an XY grid

The solid black hexagon indicates the actual polygon. For every layer the motif points are moved by a certain distance. For this example, the way in which point B will

be moved to different positions is shown. The motif is replicated starting from each of these positions, and congruent copies are created horizontally. B'' , B''' are starting positions of the second and third row respectively. 4 copies of the motif are created for bottom-most & the top-most row & 5 copies of the motif are created in the middle row. The algorithm is explained as follows:

1. Initialize the parameters:

$$horizontalD = x \text{ coordinate}_A - x \text{ coordinate}_B$$

$$distanceFromOrigin = \sqrt{x \text{ coordinate}_B^2 + y \text{ coordinate}_B^2}$$

sideLength = distance between point A and point B, which is defined as

$$= \sqrt{(x \text{ coordinate}_B - x \text{ coordinate}_A)^2 + (y \text{ coordinate}_B - y \text{ coordinate}_A)^2}$$

$$verticalD = distanceFromOrigin + \left(\frac{sideLength}{2}\right)$$

2. Initialize parameters $loop_v = 0$, $loop_h = 0$
3. If $loop_v$ is less than or equal to 2, then re-assign $loop_h = 0$ and go to step 3, otherwise exit out of the algorithm.
4. If $loop_h$ is less than or equal to 3 (if $loop_v$ is even) or 4 (if $loop_v$ is odd), then go to step 5, otherwise go to step 10.
5. For the first row, i.e., $loop_v$ is 0, assign the following variables and go to step 8

$$horizontalDistanceToMove = 2 \times horizontalD \times loop_h$$

$$verticalDistanceToMove = 0$$

6. For the second row, i.e., $loop_v$ is 1, assign the following variables & go to step 8

$$horizontalDistanceToMove = 2 \times horizontalD \times loop_h$$

$$verticalDistanceToMove = verticalD \times loop_v$$

7. For the third row, i.e., $loop_v$ is 2, assign the following variables and go to step 8

$$horizontalDistanceToMove = 2 \times (horizontalD - 1) \times loop_h$$

$$verticalDistanceToMove = verticalD \times loop_v$$

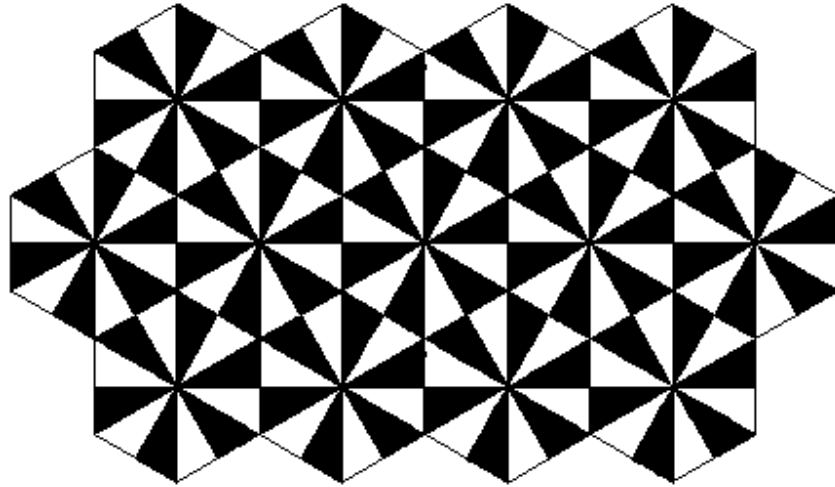
8. Redraw the Euclidean pattern by moving the original (x, y) values of the motif points to (x_{move}, y_{move}) as follows:

$$x_{move} = x + horizontalDistanceToMove$$

$$y_{move} = y + verticalDistanceToMove$$

9. Increment the value of $loop_h$ by 1, then go to step 4.
10. Increment the value of $loop_v$ by 1, then go to step 3.

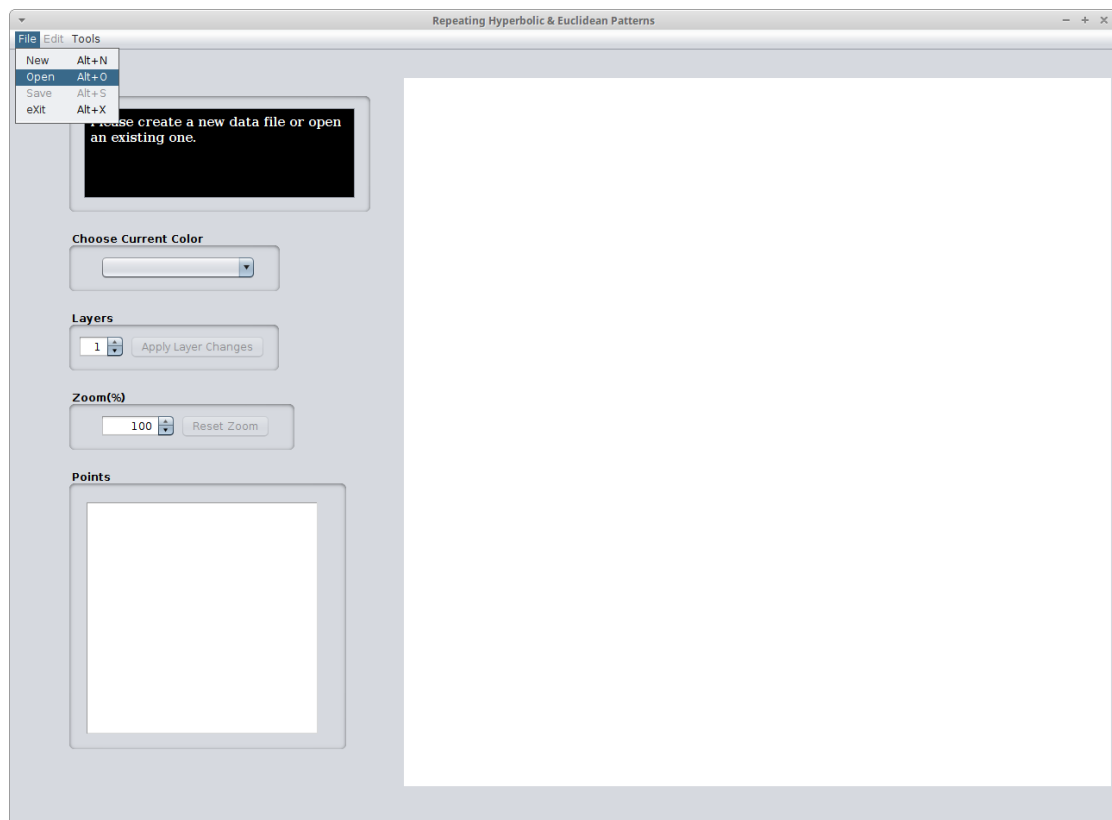
Below is a tiling pattern created for a $\{6, 3\}$ hexagon:



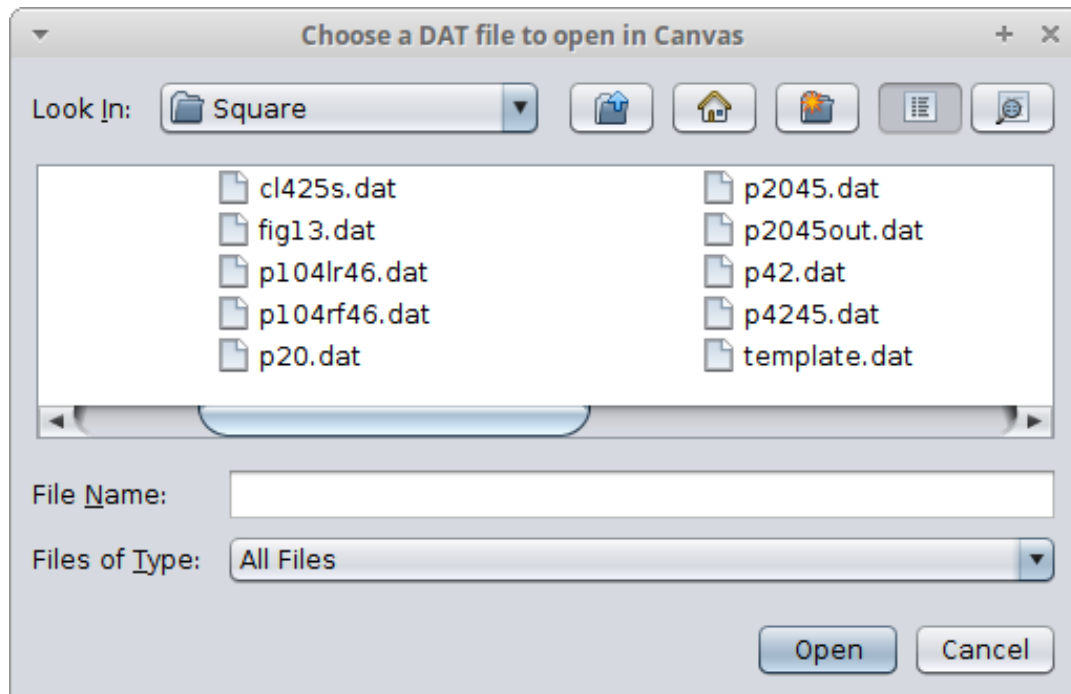
5.22 Tiling a $\{6, 3\}$ Euclidean hexagon.

6 Graphical User Interface

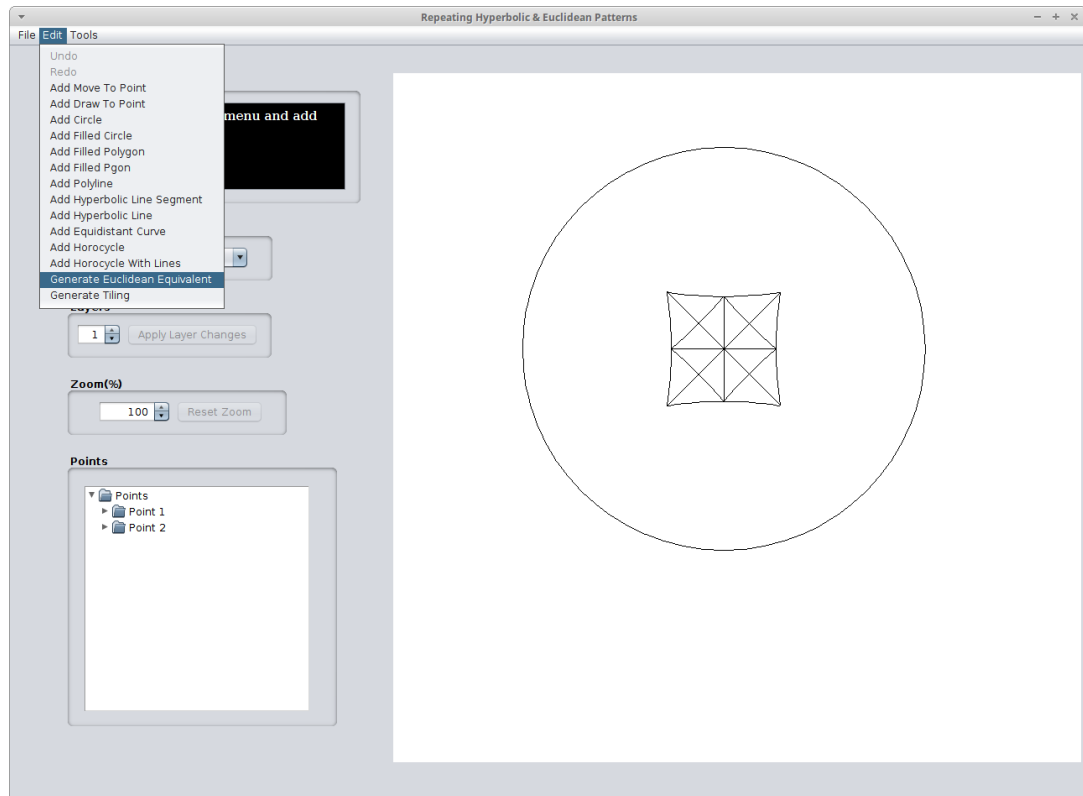
In this section describe we describe the application that allows the user to load a data file and see the Hyperbolic to Euclidean transformations and perform tilings with it. The next few screenshots are for a walk through with the basic and newly added features.



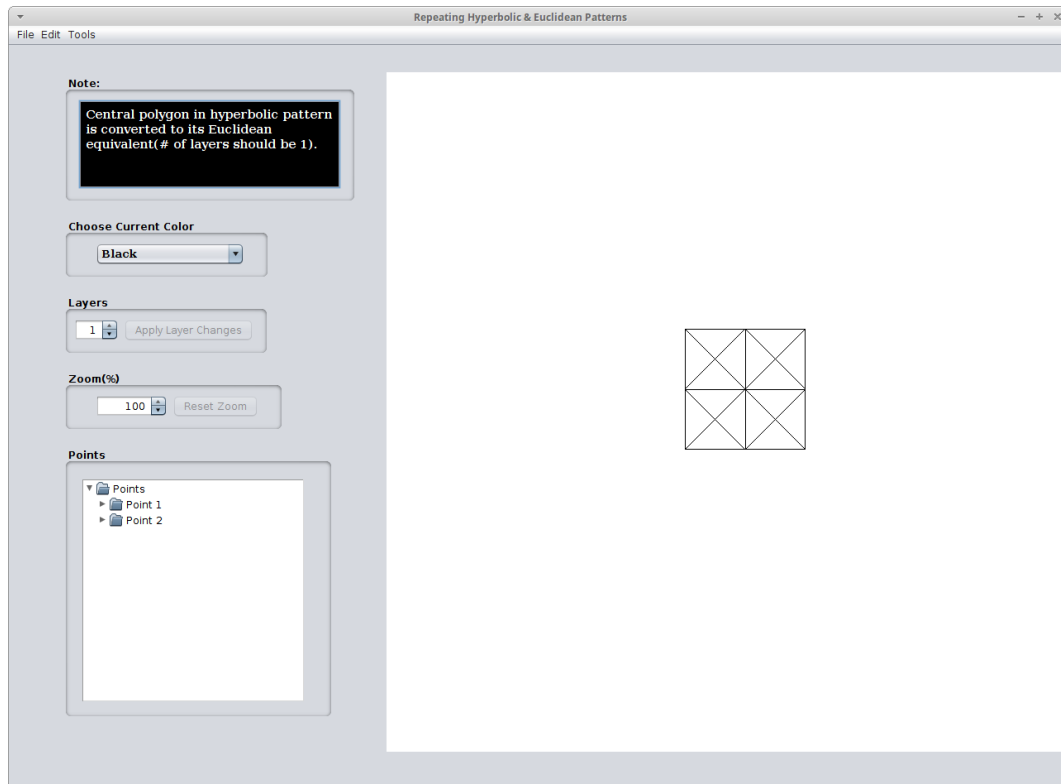
6.1 Opening a data file



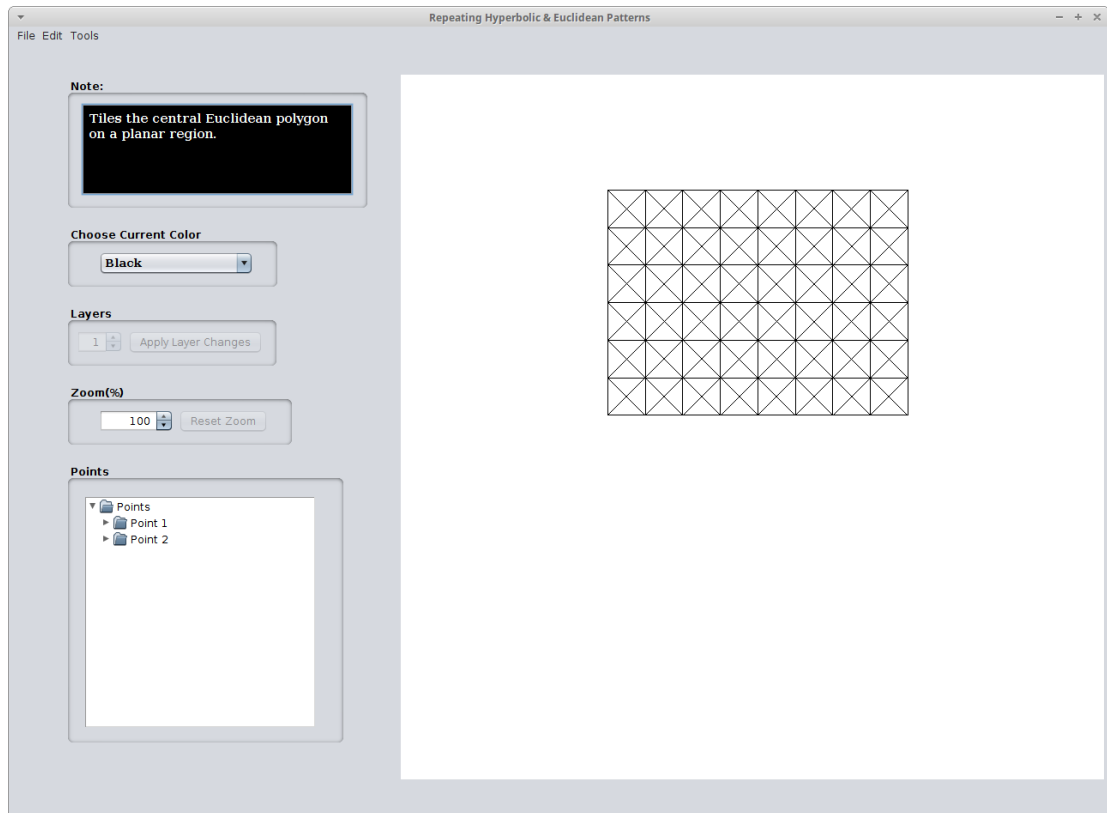
6.2 The above selection prompts a dialog box for the user to choose a data file



6.3 "Generate Euclidean Equivalent" & "Generate tiling" features under "Edit" menu section
(Hyperbolic pattern drawn on the canvas after the file is loaded)



6.4 Clicking on "Generate Euclidean Equivalent" after loading the file



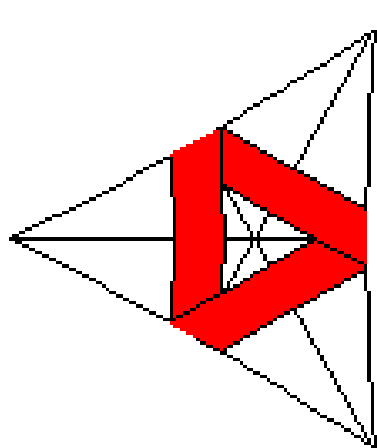
6.5 Clicking on "Generate Tiling" after loading the file

7 Results

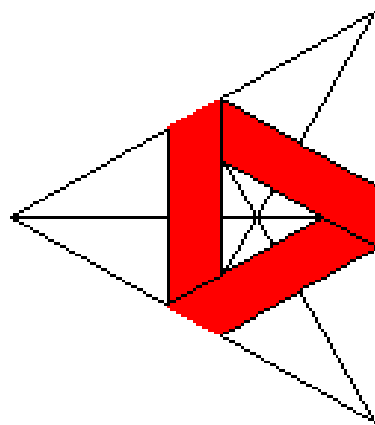
In this chapter, we present the results generated by the enhanced program. The previous applications that were developed in C++ with Qt library by Christopher Becker [13] and in Java by Maneesha Veejendla [3], show objects only in Poincaré disk model. The current program, which is an enhancement of the Java application [3] that overcomes this shortcoming by implementing features that allow the user to see the Euclidean counterparts of their hyperbolic equivalent and furthermore allows the user to create various tiling patterns.

The algorithm involved in the creation of hyperbolic central pattern is the same as described in [Section 5.2.](#), where we mentioned that central Euclidean pattern is drawn by transforming the Poincaré points to the Klein model. However for creating the hyperbolic central pattern, the Poincaré points are converted to Weierstrass model for calculations, and then corresponding objects are displayed in the Poincaré disk model (within a unit circle). The next few pages show a set of three images for every object:

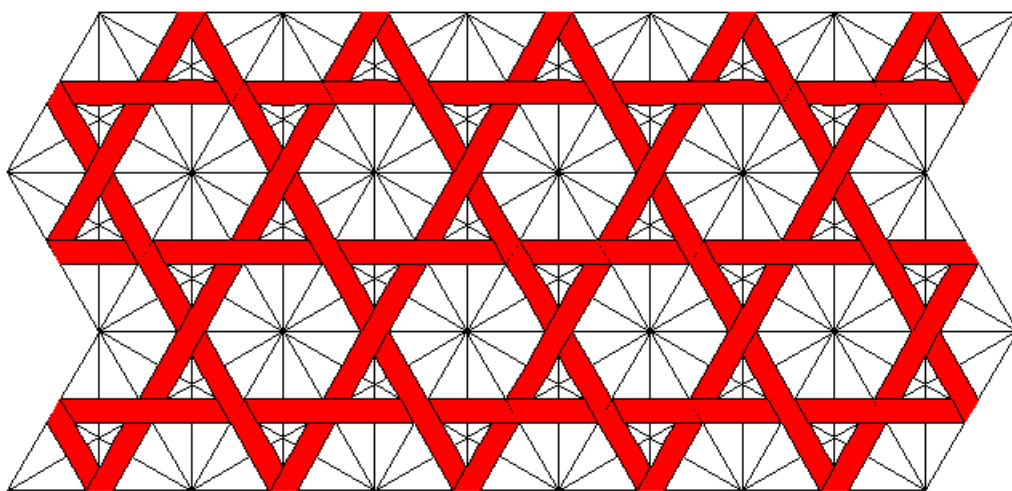
- i. The first image shows the central hyperbolic pattern.
- ii. The second image shows its equivalent Euclidean pattern.
- iii. The third image shows the final step of creating the tiling pattern of the Euclidean polygons on a planar region.



(a)

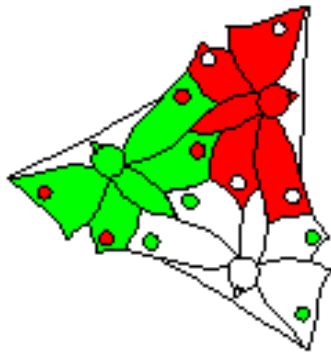


(b)

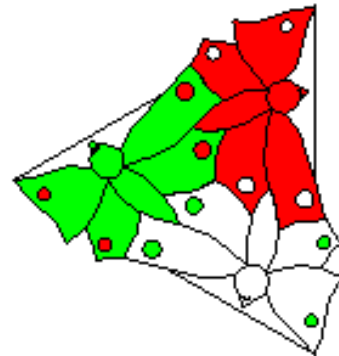


(c)

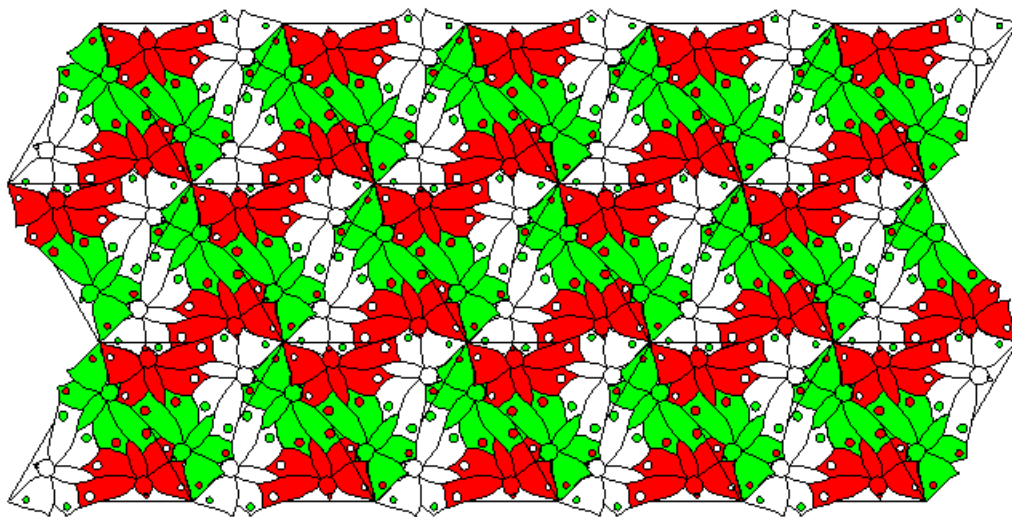
7.1 {3,7} Triangle (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern



(a)

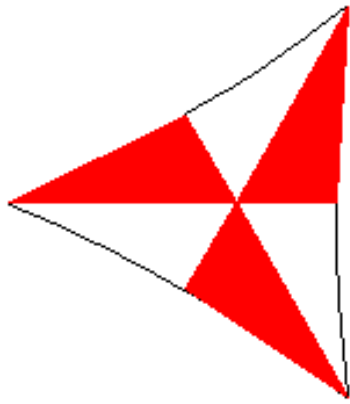


(b)

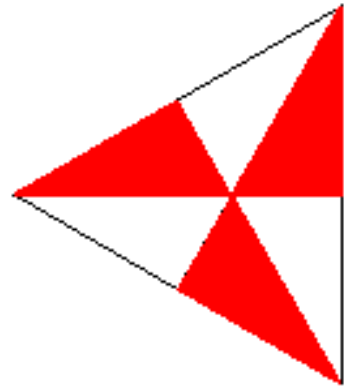


(c)

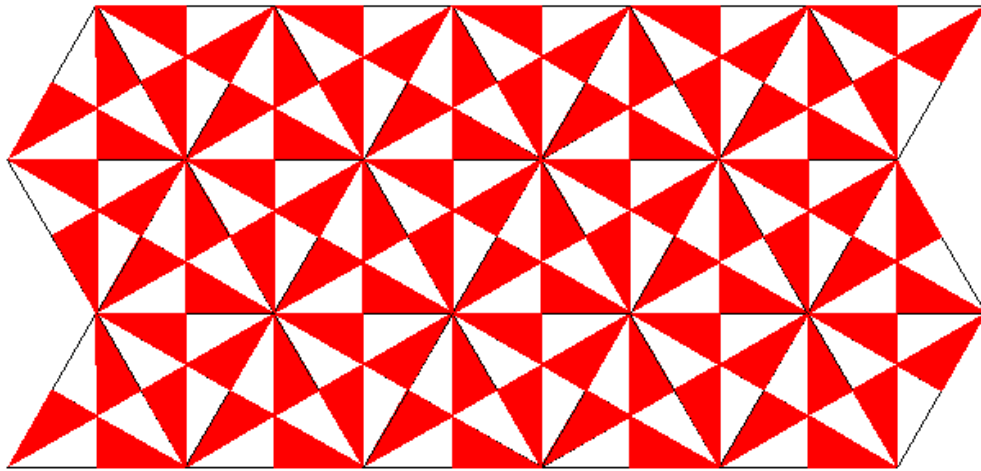
7.2 {3,7} Butterfly triangle : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern



(a)

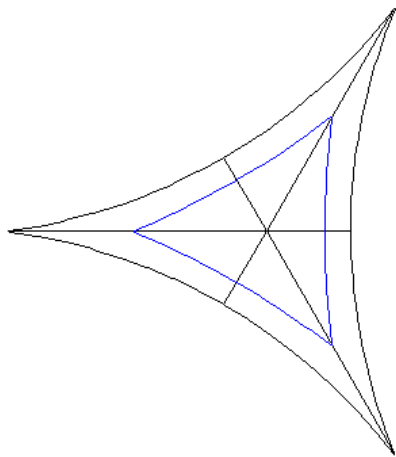


(b)

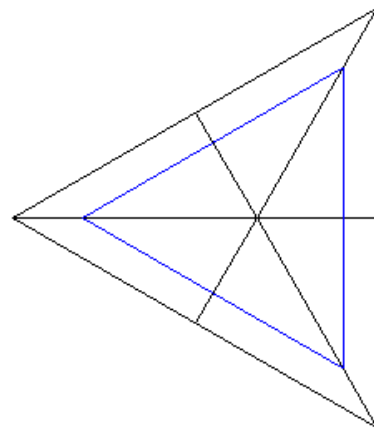


(c)

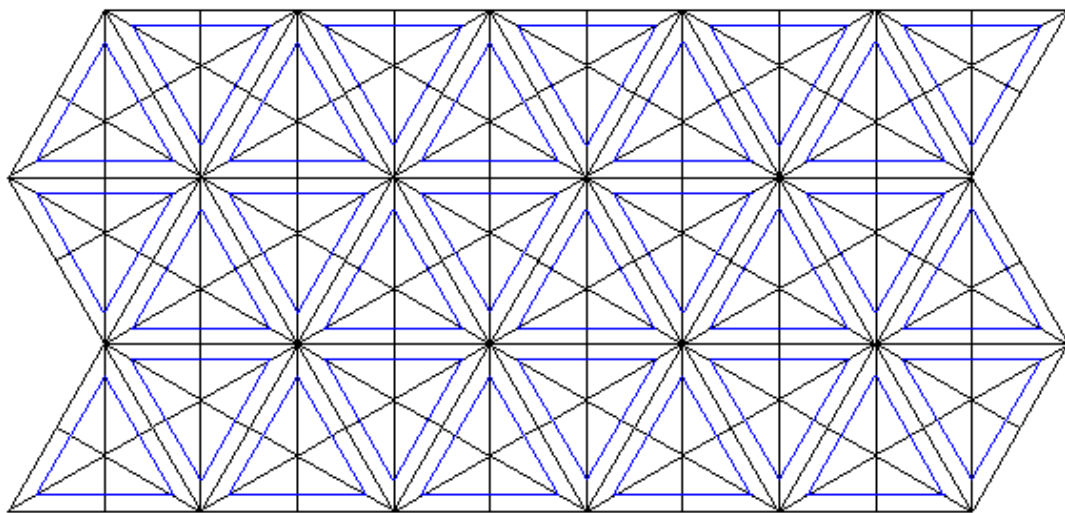
7.3 $\{3,8\}$ triangle : (a) In hyperbolic form (b) Euclidean counterpart (c) tiling pattern



(a)

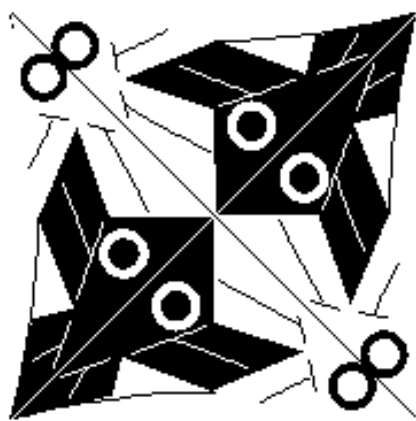


(b)

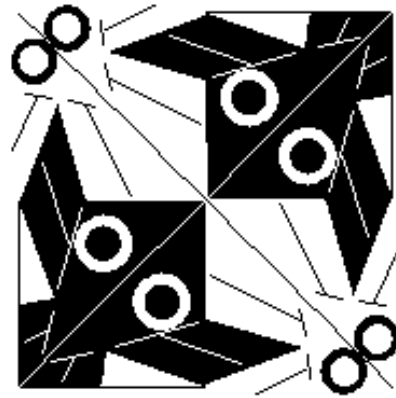


(c)

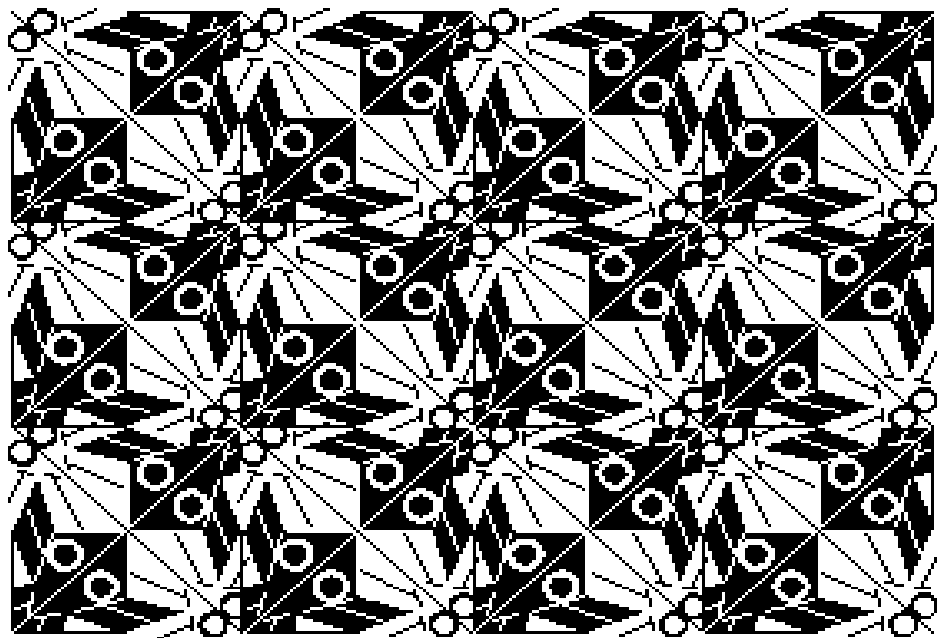
7.4 $\{3, 24\}$ triangle : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern



(a)



(b)



7.5 Escher circle limit I $\{4, 6\}$ square : (a) in hyperbolic form (b) Euclidean counterpart

(c) tiling pattern

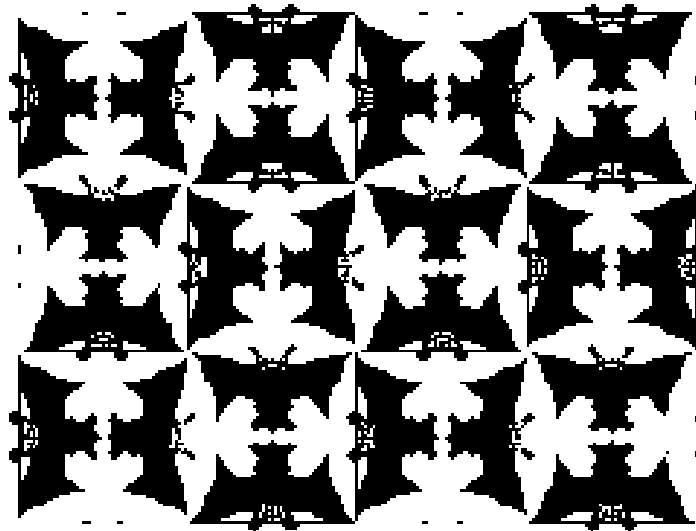


(a)



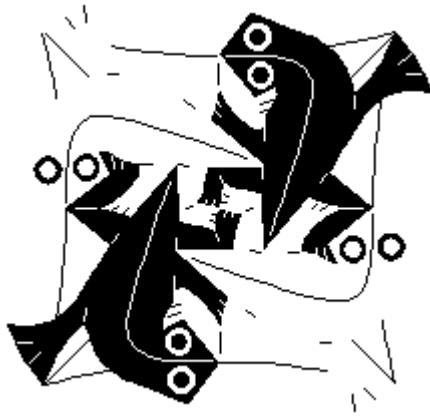
(b)

This pattern is a special case wherein the square is rotated clockwise and counter-clockwise by 90° for every alternate position.

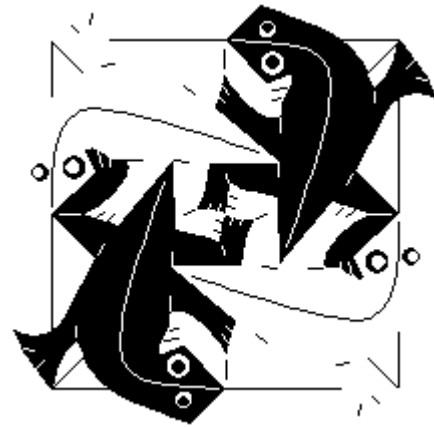


c)

7.6 A special case: Escher circle limit IV $\{4, 5\}$ square : (a) in hyperbolic form
(b) Euclidean counterpart (c) tiling pattern



(a)



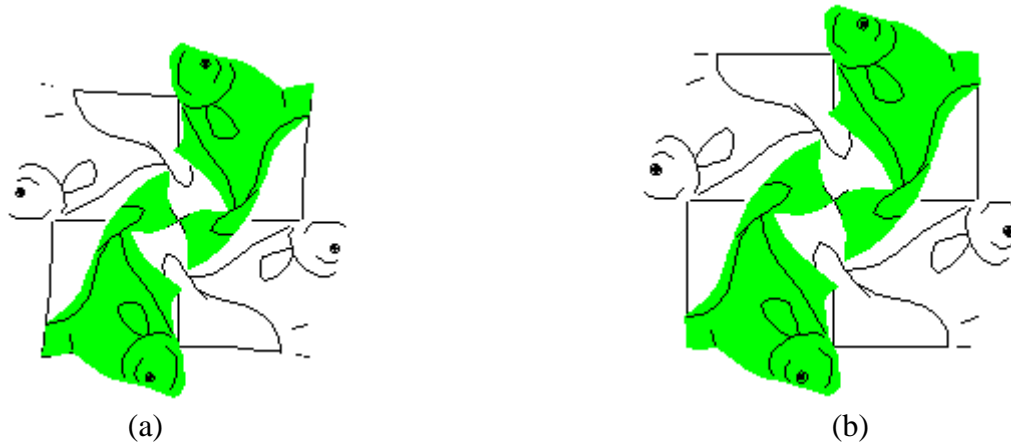
(b)



(c)

7.7 $\{4, 6\}$ frog square : (a) in hyperbolic form (b) Euclidean counterpart

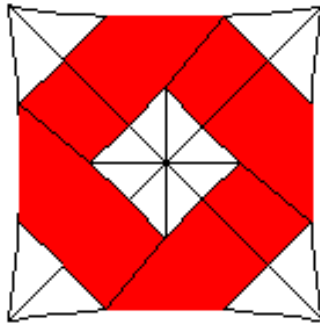
(c) tiling pattern



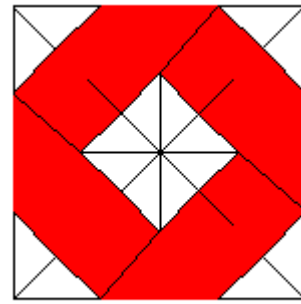
(c)

7.8 $\{4, 5\}$ fish square : (a) in hyperbolic form (b) Euclidean counterpart

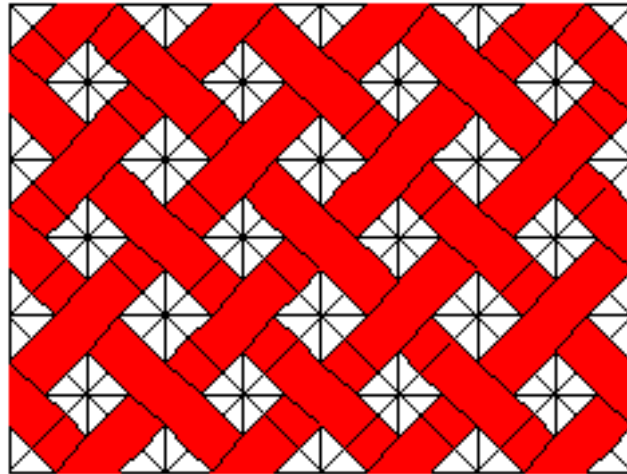
(c) tiling pattern



(a)



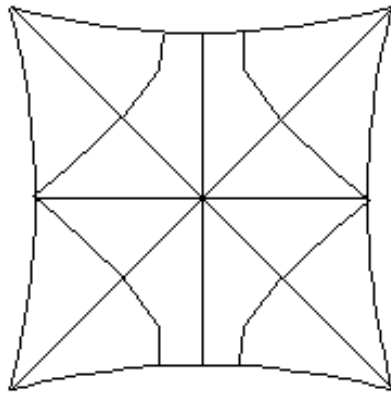
(b)



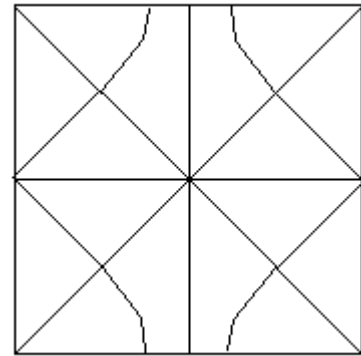
(c)

7.9 $\{4, 5\}$ square : (a) in hyperbolic form (b) Euclidean counterpart

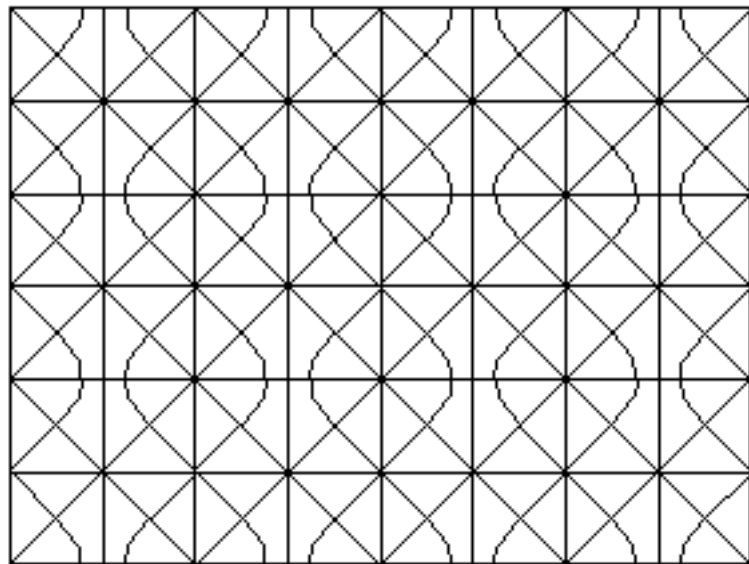
(c) tiling pattern



(a)

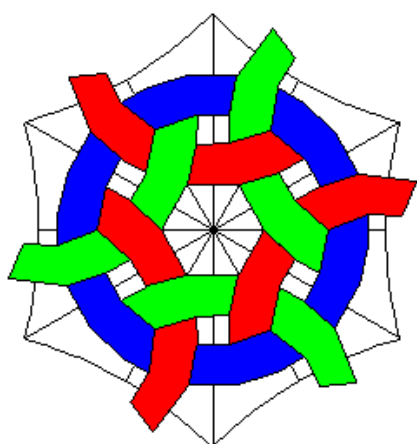


(b)

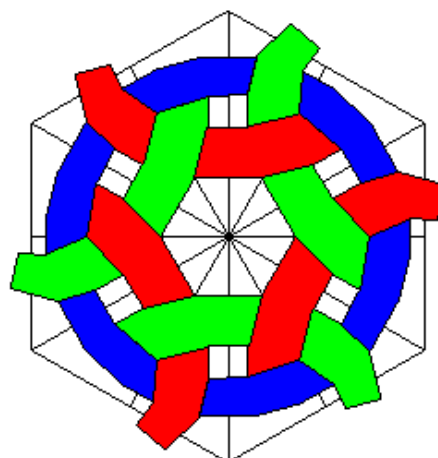


(c)

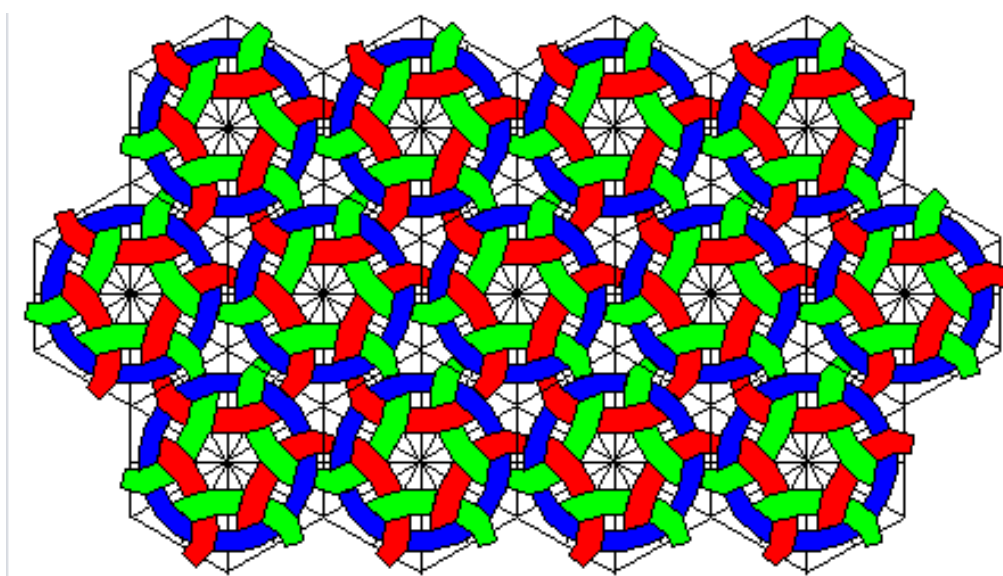
7.10 $\{4, 6\}$ square : (a) in hyperbolic form (b) Euclidean counterpart
(c) tiling pattern



(a)

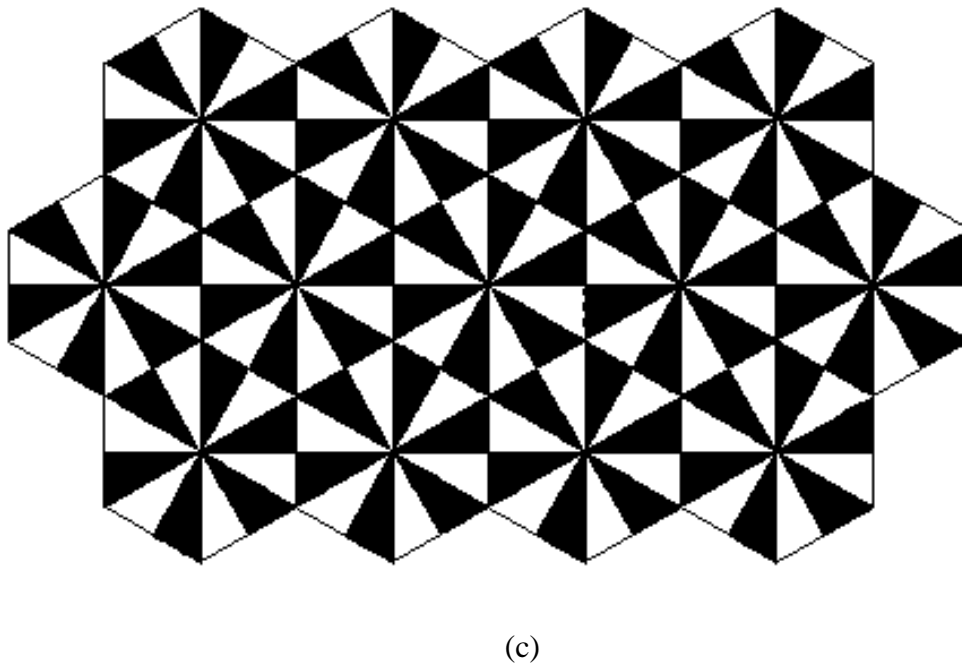
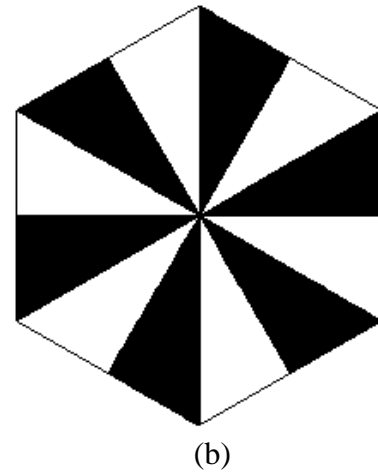
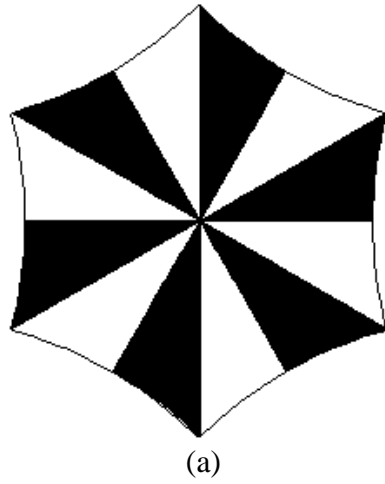


(b)

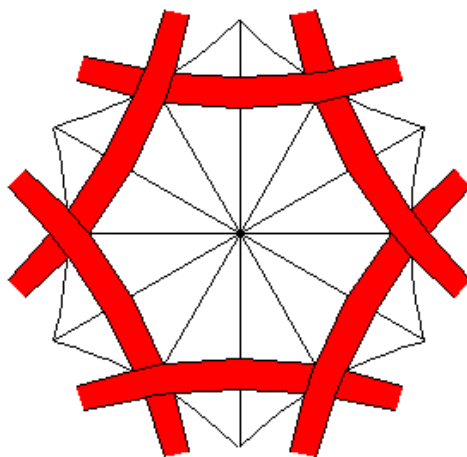


(c)

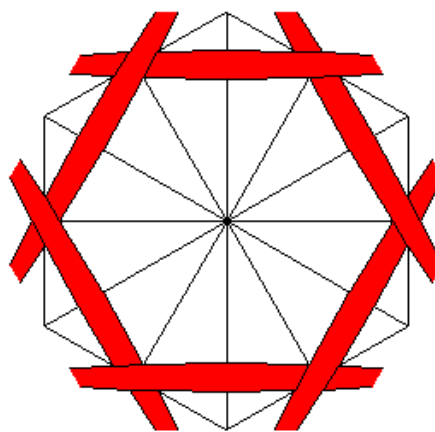
7.11 $\{6, 4\}$ hexagon with woven ribbons : (a) in hyperbolic form (b) Euclidean counterpart
(c) tiling pattern



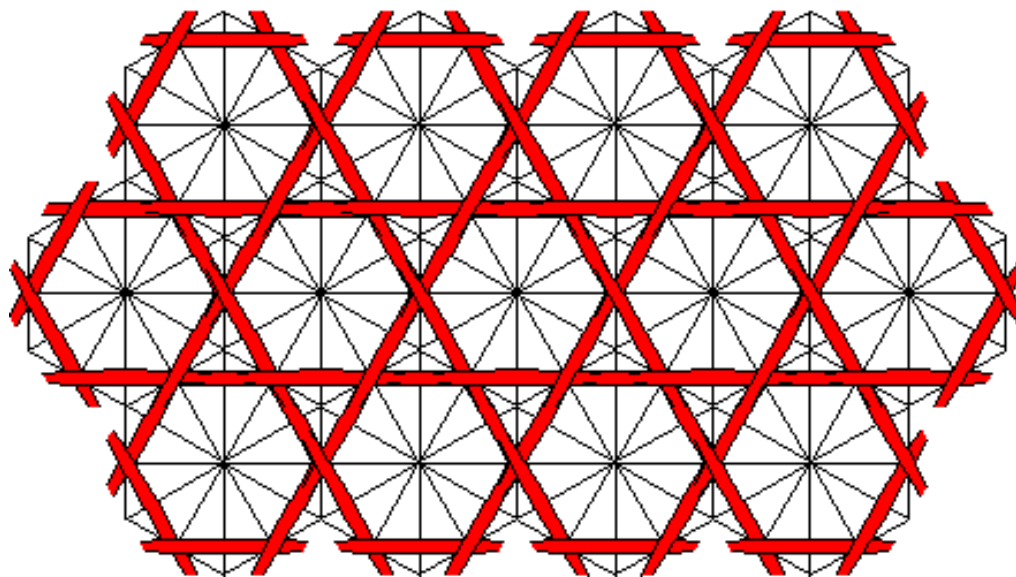
7.12 Central polygon in Coxeter's figure - $\{6, 4\}$ hexagon : (a) in hyperbolic form
(b) Euclidean counterpart (c) tiling pattern



(a)



(b)

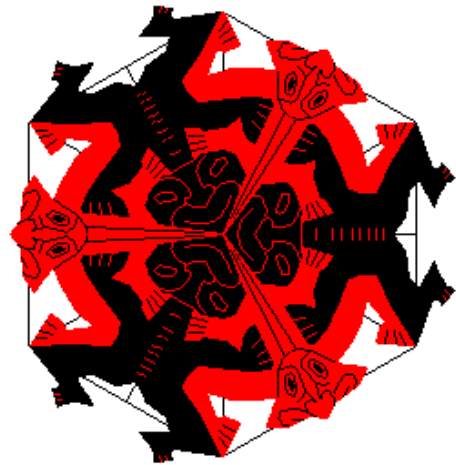


(c)

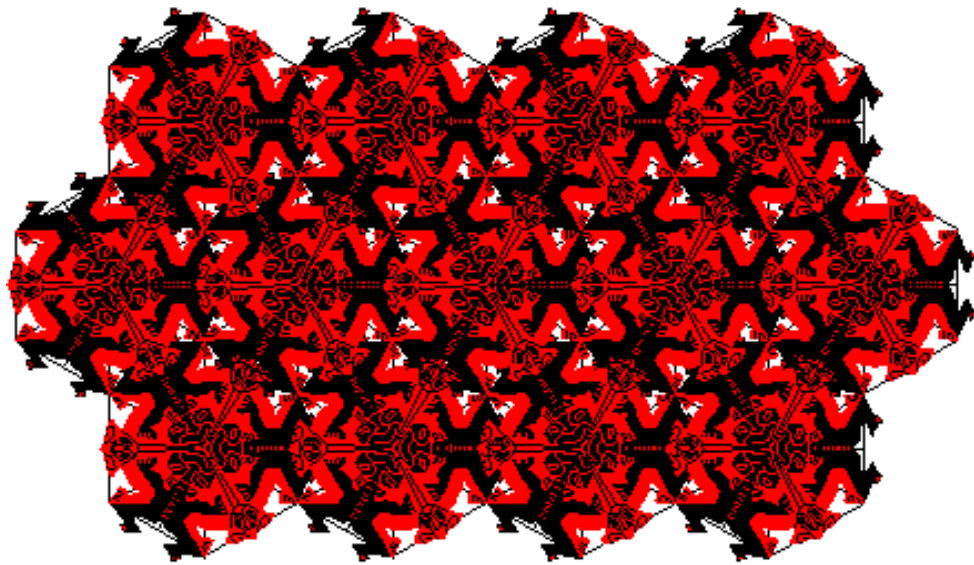
7.13 A $\{6, 4\}$ hexagon with scattered strips of ribbon: (a) in hyperbolic form
 (b) Euclidean counterpart (c) tiling pattern



(a)



(b)



(c)

7.14 $\{6, 4\}$ hexagon : (a) in hyperbolic form (b) Euclidean counterpart (c) tiling pattern

8 Conclusion

The research focus aims to depict Euclidean objects using the Klein model of the hyperbolic geometry. Copies of central Euclidean region are made to create a perfect tiling on a planar region. Until now, most of the work has been done in Weierstrass model of hyperbolic geometry; using such a tiling might create holes or gaps, which is undesirable. We have extended the features of an existing Java application [3] for the user to see a mapping between objects in the Poincaré model and the Klein model, and eventually create different tilings that perfectly fit into a Euclidean region without any gaps in between. The program has been tested rigorously with different data files for triangles, squares and hexagons.

9 Future Work

The goal of this research work was to use regular Euclidean polygons that perfectly tile a planar region. The tilings are generated using equilateral triangles, or squares, or regular hexagons. This work could be extended to create semi-regular tilings using regular polygons. Also another direction of research would be to create tilings using non-regular polygons.

Currently the application allows the user to load the file and create tilings of fixed size. This can be extended to allow the user to generate the tilings of different size, and also let the user rotate the final product horizontally or vertically.

References

- [1] Dunham, Douglas and Lindgren, John and Witte, David, "Creating Repeating Hyperbolic Patterns", Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques.
{<http://doi.acm.org/10.1145/800224.806808>}
- [2] Dunham, Douglas, "How to Create Repeating Hyperbolic Patterns"
{<https://www.d.umn.edu/~ddunham/eccad09.pdf>}
- [3] Vejendla, Maneesha, "Generating Repeating Hyperbolic Patterns based on Regular Tessellation using an Applet"
http://www.d.umn.edu/cs/thesis/Vejendla_Maneesha_December2013.pdf
- [4] N.-E. Geometry. "H. S. M. Coxeter"
{http://www.aproged.pt/biblioteca/geometryrevisited_coxetergreitzer.pdf}
- [5] Alexander Bogomolny, *Non-Euclidean Geometries, Discovery from Interactive Mathematics Miscellany and Puzzles*
<http://www.cut-the-knot.org/triangle/pythpar/Drama.shtml>, Accessed 22 April 2017
- [6] James W. Cannon, William J. Floyd, Richard Kenyon, and Walter R. Parry, "Hyperbolic Geometry"
{<https://www.math.brown.edu/~rkenyon/papers/cannon.pdf>}
- [7] Weisstein, Eric W. "Elliptic Geometry." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/EllipticGeometry.html>
- [8] J. W., Anderson, "The Poincaré Disc Model." in "Hyperbolic Geometry"
- [9] S. Karumuri, "An Interactive JAVA Program to Generate Hyperbolic Repeating Patterns Based on Regular Tessellations Including Hyperbolic Lines Equidistant Curves". MA thesis. UNIVERSITY OF MINNESOTA, 2015
- [10] Richard L., Faber, "Foundations of Euclidean and Non - Euclidean Geometry"
- [11] Dunham, Douglas, " USE OF MODELS OF HYPERBOLIC GEOMETRY IN THE CREATION OF HYPERBOLIC PATTERNS"
- [12] Data, Ajit, " GENERATING HYPERBOLIC PATTERNS FOR REGULAR AND NON-REGULAR P-GONS",
http://www.d.umn.edu/cs/thesis/ajit_datar_ms.pdf
- [13] Christopher D. Becker. Creating Repeating Hyperbolic Patterns Based on Regular Tessellations. Master's Thesis, University of Minnesota Duluth 2012.
- [14] https://en.wikipedia.org/wiki/Wallpaper_group

Appendix A

Sample Data File Format

The following section contains line by line explanation of the data file (*tri46fil.dat*) in reference to the [Figure 1.1](#). The data file contents are:

4 6 1 0 3 0

1 2 3

1 2 3

1 1 2 3

2 1 2 3

3 1 2 3

4 1 2 3

4

0.000000e+00 0.000000e+00 1 12 3

3.178372e-01 0.000000e+00 1 13 3

3.660254e-01 3.660254e-01 1 13 3

0.000000e+00 0.000000e+00 1 14 3

The first line : 4 6 1 0 3 0 is explained as follows:

1. The first number is the value of p . Here it is a 4 sided polygon.
2. The second number is the value of q , which indicates the type of tessellations{p, q}
3. The third number (say s) : 1 indicated the number of sides in the central region

i.e., p divided by this number is the total number of copies to be made of the motif, to create the central polygon. In this case, there are 4 copies of the motif made.

4. The fourth number is not used at all, it is only used to maintain consistency with the previous applications.
5. The fifth number, 3, is the highest color number of the colors to be used. The different color numbers are:

- 1 Black
- 2 White
- 3 Red
- 4 Green
- 5 Blue
- 6 Cyan
- 7 Magenta
- 8 Yellow
- 9 Salmon
- 10 Brown

6. The last number, 0, indicates the type of reflection symmetry the pattern has with respect to the central p -sided polygon. There are three possible values:

- 0 denotes there is only rotation symmetry and no reflection symmetry.
- 1 denotes there is a reflection symmetry across the perpendicular bisector of one of the edges of the p -sided polygon.

- 2 indicates there is a reflection symmetry across the radius.

The second line : 1 2 3, is the color permutation induced by rotation.

The third line : 1 2 3, is the color permutation induced by reflection (if the number is 0 on line 1, it is just the identity).

The next p lines (4 in this case) consist a number followed by a color permutation.

The first number of the first line represents the edge (edge 1 in this case) of the transformed pgon that should lie next to edge 1 of the central pgon. Generally, if the first number is positive, then the transformed pgon is rotated into position, otherwise if it's negative, a reflection is used to move the transformed pgon into position. The edges are numbered 1 to p .

The values following the first number, 1 2 3 in this case, indicates the color permutation.

The next line: a single number (2), denotes the total number of (x, y) points that generate the motif.

The next 2 lines has the below structure with 5 space separated components:

<x-coordinate> <y-coordinate> <color> <type of point> <number of layers>

The above line is explained as follows:

1. The x-coordinate & y-coordinate within the central polygon.
2. Color is the number, explained earlier.
3. Type of point is of the following:
 - 1 "Move To"
 - 2 "Draw To"
 - 3 "Circle" (these should come in pair i.e., the next point type should also be 3)

- 4 Start a Euclidean Filled Polygon
- 5 Continue a Euclidean Filled Polygon
- 6 Finish a Euclidean Filled Polygon
- 7 Hyperline (these should come in pair i.e., the next point type should also be 7)
- 8 Filled Circle (these should come in pair i.e., the next point type should also be 8)
- 9 Start a Euclidean Polyline
- 10 Continue a Euclidean Polyline
- 11 Finish a Euclidean Polyline
- 12 Start a hyperbolic filled p-gon
- 13 Continue a hyperbolic filled p-gon
- 14 End a hyperbolic filled p-gon

The number of layers value is not used, it only exists to maintain reverse compatibility with the old application.